

An Optimal Algorithm for Strict Circular Seriation*

Santiago Armstrong[†], Cristóbal Guzmán^{†‡}, and Carlos A. Sing Long^{†§}

Abstract. We study the problem of circular seriation, where we are given a matrix of pairwise dissimilarities between n objects and the goal is to find a *circular order* of the objects in a manner that is consistent with their dissimilarity. This problem is a generalization of the classical *linear seriation* problem, where the goal is to find a *linear order* and for which optimal $\mathcal{O}(n^2)$ algorithms are known. Our contributions can be summarized as follows. First, we introduce *circular Robinson matrices* as the natural class of dissimilarity matrices for the circular seriation problem. Second, for the case of *strict circular Robinson dissimilarity matrices* we provide an optimal $\mathcal{O}(n^2)$ algorithm for the circular seriation problem. Finally, we propose a statistical model to analyze the well-posedness of the circular seriation problem for large n . In particular, we establish $\mathcal{O}(\log(n)/n)$ rates on the distance between any circular ordering found by solving the circular seriation problem to the underlying order of the model in the Kendall-tau metric.

Key words. circular seriation, circular Robinson dissimilarities, PQ -trees, circular Robinsonian matrices, circular-arc hypergraphs, circular embeddings of graphs, generative model

AMS subject classifications. 68R01, 05C85, 05C50, 05C25, 65C20

DOI. 10.1137/21M139356X

1. Introduction. The seriation problem seeks to order a sequence of n objects from pairwise dissimilarity information. The goal is for the objects to be linearly ordered according to their dissimilarity [17, 25, 26]. Seriation has found applications in several areas such as archaeology [27], sociology and psychology [17], and gene sequencing and bioinformatics [25, 19]. However, in many applications the objects may be arranged along a closed continuum, resulting in a circular order instead. For instance, in de novo genome assembly of bacterial plasmids, the goal is to reorder DNA fragments sampled from a circular genome [25, 16]. In some problems in planar tomography, an object's density is to be reconstructed from projections taken at unknown angles between 0 and 2π . Reordering the projections according to their angle enables the reconstruction of the density [9]. In this case, the matrix representation of the

*Received by the editors January 21, 2021; accepted for publication (in revised form) August 11, 2021; published electronically November 11, 2021.

<https://doi.org/10.1137/21M139356X>

Funding: This work was partially supported by INRIA through the INRIA Associate Teams project, by CORFO through the Clover 2030 Engineering Strategy (14ENI-26862), and by ANID – Millennium Science Initiative Program (NCN17_059). The third author was partially supported by ANID – Millennium Science Initiative Program (NCN17_129).

[†]Institute for Mathematical and Computational Engineering, Pontificia Universidad Católica de Chile, Santiago, Chile and ANID – Millennium Science Initiative Program – Millennium Nucleus Center for the Discovery of Structures in Complex Data, Santiago, Chile (sarmstrong@uc.cl, casinglo@uc.cl).

[‡]Department of Applied Mathematics, University of Twente, Enschede, The Netherlands (c.guzman@utwente.nl).

[§]Institute for Biological and Medical Engineering, Pontificia Universidad Católica de Chile, Santiago, Chile and ANID – Millennium Science Initiative Program – Millennium Nucleus Center for Cardiovascular Magnetic Resonance, Santiago, Chile.

pairwise dissimilarities is symmetric, with entries that increase monotonically starting from the diagonal along each row until they reach a maximum and then decrease monotonically, when the columns are wrapped around (see Figure 1). Matrices of this form are called circular Robinson [10, 12] in contrast to linear Robinson dissimilarities, where the entries are monotone nondecreasing along rows and columns when moving toward the diagonal [15].

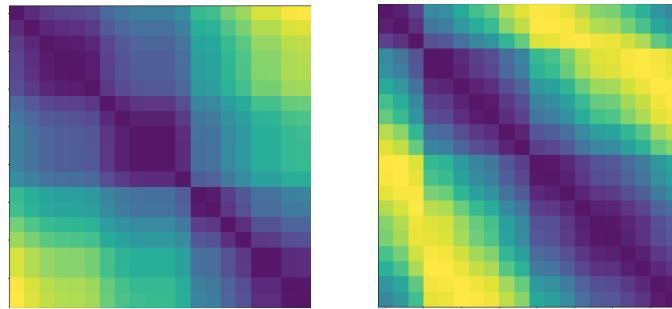


Figure 1. Example of a linear Robinson dissimilarity matrix (in the left) and a circular Robinson dissimilarity matrix (in the right).

1.1. Our contributions. In this work, we address the problem of circular seriation. In particular, we address algorithmic questions and study its well-posedness for large n . Some of our results also apply to the linear case. Our first contribution is to provide a tractable and natural definition of circular Robinson matrices by leveraging unimodality (cf. Proposition 3.7). Various definitions of circular ordering have been proposed in the literature (see subsection 1.2 below), but we believe this one captures intuitively the behavior of circular data.

Our second contribution is to provide the first optimal algorithm, i.e., with $\mathcal{O}(n^2)$ time and space complexity, for the seriation problem on strict Robinson dissimilarity matrices. Our algorithm is based on known techniques and data structures used in combinatorial seriation, but by virtue of the strict Robinson property our algorithm is substantially simpler. At a high level, the algorithm follows a divide-and-conquer approach, where we recursively detect nearest neighbors between chains of consecutive elements and then resolve the orientations of such chains by comparing elements from their borders.

Our third contribution is a statistical model for the large n regime. In this model, points are sampled from a closed curve, which without loss of generality we assume is the unit circle, with a continuous and strict circular Robinson dissimilarity. Our main result here is an $\mathcal{O}(\log(n)/n)$ bound on the expected Kendall-tau distance of any strict circular Robinson ordering of the data. This result is based on an observation that in the continuous model, there is essentially¹ a unique ordering which makes the dissimilarity continuous and strictly circular Robinson. This analysis bridges the gap between solutions to the seriation problem and their accuracy when data is naturally embedded in a continuous circular-like structure.

¹In an infinite set, permutations can be identified with bijections. However, given that for any finite sample we would only observe permutations of finitely many elements, we can substantially reduce the number of relevant permutations for this question. See section 6 for further details.

1.2. Related work. Linear seriation is a classical problem in unsupervised learning and exploratory data analysis. As such, it has been thoroughly studied; optimal algorithms for combinatorial seriation are known, as well as spectral methods. In contrast, circular seriation is substantially less understood. Next we summarize some results from the literature.

Linear seriation. The first polynomial time algorithm for retrieving a linear order from permuted linear Robinson matrices was due to Mirkin and Rodin [19]. It is based on the connection between linear Robinson matrices and interval hypergraphs. It uses an algorithm introduced in [11] as a core subroutine, with an overall running time of $\mathcal{O}(n^4)$. Chepoi and Fichet [6] later introduced a simpler algorithm using a divide-and-conquer strategy. By recursively performing a partition refinement the algorithm computes an ordering in $\mathcal{O}(n^3)$ operations and $\mathcal{O}(n^2)$ space. Using similar techniques, Seston [28] improved the complexity to $\mathcal{O}(n^2 \log(n))$. Atkins, Boman, and Hendrickson [1] presented an entirely different strategy based on Laplacian eigenmaps (see [2]) with running time of $\mathcal{O}(n(T(n) + n \log n))$, where $T(n)$ is the complexity of (approximately) computing the leading eigenvector of a $n \times n$ symmetric matrix. Pr ea and Fortin in [22] presented an optimal $\mathcal{O}(n^2)$ algorithm, using an algorithm from [4] to first compute a PQ -tree which is then updated by the algorithm. For the sparse case, Laurent and Seminaroti [15] present the similarity-first search algorithm with $\mathcal{O}(n^2 + nm \log n)$ operations, where m is the number of nonzero entries of the dissimilarity matrix.

A natural question is how to perform seriation under noisy measurements of a dissimilarity. Here, it is known that projecting a dissimilarity on the class of Robinsonian dissimilarities (in ℓ_∞ -norm) is an NP-hard problem [7], and constant factor approximation algorithms exist [8].

Circular seriation. In contrast to the linear case, where there is a common consensus for the definition of linear Robinson dissimilarities, in the circular case many definitions have been proposed that, although they follow the same intuition, have mathematical formulations that are not equivalent. The first generalization of Robinson dissimilarities to the circular case was introduced in [12]. On top of being quite involved, this definition allows bimodality within each row (modulo n), which is incompatible with a circle embedding. The approach proposed for circular seriation is an instance of the quadratic assignment problem, which is NP-hard. A recent work following a similar line is [10]. The authors propose an optimization framework where they employ a spherical embedding together with a spectral method for circular ordering in order to recover circular arrangements of the embedded objects. This heuristic has no theoretical guarantees. A different approach in [9] aims to generalize Atkins' spectral approach by considering two eigenvectors. This methodology has asymptotic guarantees due to the connection between the Laplacian operator and the continuous Laplace–Beltrami operator over a manifold. Using the same idea, in [26] theoretical guarantees for a spectral method are introduced for the particular case in which the circular Robinson matrix is circulant, which is an idealized setting. In the same work, numerical experiments are presented to illustrate how the spectral method gains robustness by leveraging higher (>2) Laplacian eigenvectors. In [5] dissimilarities whose ball, 2-ball, and cluster hypergraph correspond to an arc hypergraph are studied. Such dissimilarities can be considered as generalizations of Robinson dissimilarities to the circular case. We build upon this work by considering dissimilarities whose ball hypergraph corresponds to arcs and connect it to other definitions by showing that this definition is equivalent to requiring that the map $j \mapsto D(i, j + i \bmod n)$ is unimodal. Brucker and Osswald

in [5] mainly focus in what they call circular dissimilarities which are a particular case of the previous definition.

1.3. Outline. The paper is organized as follows. [Section 2](#) introduces the notation and preliminaries. In [section 3](#) we formally introduce the seriation problem and the crucial concept of Robinson dissimilarities and matrices. In [section 4](#) we present some classical results on the *consecutive ones problem* and its connection to seriation, including the *PQ-tree data structure*, which is critical for our optimal algorithm. In [section 5](#) we present our optimal algorithm for strict circular seriation. Finally, in [section 6](#) we provide the generative model of sampling from a *continuous strictly Robinson curve*.

2. Preliminaries. Throughout this work, arrays are indexed starting from 0 and are real unless it is explicitly stated otherwise. We let $[n] \triangleq \{0, 1, \dots, n-1\}$ and denote as $\text{Sym}(n)$ the group of permutations of $[n]$. A permutation is represented either by a vector π with entries in $[n]$ or by an $n \times n$ orthogonal and $\{0, 1\}$ -matrix Π . We denote as π_r the permutation that *reverses* the elements of $[n]$, i.e., $\pi_r(i) = n - 1 - i$, and π_s the *cyclic (right) shift* on $[n]$, i.e., $\pi_s(i) = i + 1 \pmod n$. We consider the *action by conjugation* of $\text{Sym}(n)$ over the set of $n \times n$ matrices, which is defined by $(\Pi, A) \mapsto \Pi A \Pi^T$. If $S \subset \text{Sym}(n)$ we denote $\langle S \rangle$ the subgroup generated by the elements of S . Finally, we denote the dihedral group of $2n$ different symmetries of a regular polygon with n sides as Dih_n .

For a countable set \mathcal{X} and an enumeration $x : i \rightarrow x(i)$ we write x_i to denote $x(i)$ and let $\#x$ be the integer such that $x(\#x) = x$. In this work, we consider finite sets of cardinality n . An enumeration becomes a bijection $[n] \mapsto \mathcal{X}$ with inverse $\# : \mathcal{X} \rightarrow [n]$.

The notion of an *ordered set* will play a crucial role. A *linear order* on \mathcal{X} is a relation \leq on \mathcal{X}^2 that is reflexive, antisymmetric, transitive, and total. The pair (\mathcal{X}, \leq) is a *linearly ordered set*. We say x_0, \dots, x_{N-1} are *linearly ordered* if $x_i \leq x_{i+1}$ for $i \in [N]$. A *cyclic order* on \mathcal{X} is a relation \mathcal{C} on \mathcal{X}^3 that is cyclic, antisymmetric, transitive, and total. The pair $(\mathcal{X}, \mathcal{C})$ is a *cyclically ordered set*. A cyclic order induces a linear order on \mathcal{X} . For $x_0 \in \mathcal{X}$ we define the linear order $\leq_{\mathcal{C}, x_0}$ as $x \leq_{\mathcal{C}, x_0} y$ if and only if $(x_0, x, y) \in \mathcal{C}$. Finally, we say $x_0, \dots, x_{N-1} \in \mathcal{X}$ are *cyclically ordered* if $x_i \leq_{\mathcal{C}, x_0} x_{i+1}$ for $i \in [N]$. See [20] for more details.

3. The seriation problem and Robinson dissimilarities. We introduce the seriation problem. Given a set \mathcal{M} of $n \times n$ real matrices, let the *pre- \mathcal{M} class* be the orbit of \mathcal{M} under the action of $\text{Sym}(n)$ by conjugation. The *abstract seriation problem* can be stated as [26]

Given A in pre- \mathcal{M} find Π in $\text{Sym}(n)$ such that $\Pi A \Pi^T$ is in \mathcal{M} .

The seriation problem is determined by the class \mathcal{M} . A *solution* to the seriation problem for A is any permutation Π satisfying the above. We denote the set of all solutions by $\mathcal{S}_{\mathcal{M}}(A)$.

We study two questions about this problem: *for which class \mathcal{M} can we ensure a solution exists?* and, given this class, *is there an efficient algorithm to solve the seriation problem for any A ?* In this work we focus on the case where the class \mathcal{M} is induced by a dissimilarity on a finite set \mathcal{X} . Our goal is to provide an answer when this dissimilarity may induce a linear or cyclic order on this set. For this reason, we explicitly distinguish between the *linear seriation problem* and the *circular seriation problem*; the *seriation problem* refers to either of them.

To answer the first question, in [subsection 3.1](#) we characterize dissimilarities that admit such linear or cyclic orders, and in [subsection 3.2](#) we discuss how these induce a suitable class

of matrices for the seriation problem. We defer the answer to the second question to [section 4](#) and [section 5](#).

3.1. Robinson dissimilarities. Most of the work on the seriation problem has focused on matrix dissimilarities. In this work, we primarily focus on an equivalent formulation based on abstract pairwise dissimilarities. This allows for a more transparent presentation of algorithms, as well as a more natural extension to the case of infinite continuous sets. Later, we provide a formal connection between the two concepts.

A *dissimilarity* or *premetric* $\mathbf{d} : \mathcal{X}^2 \rightarrow \mathbb{R}$ on \mathcal{X} is a nonnegative and symmetric function that is identically zero on the diagonal. *Robinson dissimilarities* are dissimilarities to which we can associate a linear or cyclic order on \mathcal{X} .

3.1.1. Linear Robinson dissimilarities. Linear Robinson dissimilarities admit a family of linear orders on \mathcal{X} .

Definition 3.1 (the linear Robinson property). A dissimilarity \mathbf{d} on \mathcal{X} is linear Robinson if there exists a linear order $\leq_{\mathbf{d}}$ on \mathcal{X} such that

$$(3.1) \quad \forall \text{ linearly ordered } x, y, z \in \mathcal{X} : \mathbf{d}(x, z) \geq \max\{\mathbf{d}(y, x), \mathbf{d}(y, z)\}.$$

It is strictly linear Robinson if all the inequalities are strict. We say $\leq_{\mathbf{d}}$ is consistent with \mathbf{d} and that \mathbf{d} is linear Robinson with respect to $\leq_{\mathbf{d}}$.

Linear Robinson dissimilarities preserve the *intervals* defined by any consistent order [19]. From [Definition 3.1](#) it follows that for any $r > 0$ and $x \in \mathcal{X}$ the (closed) balls $B_r^{\mathbf{d}}(x) \triangleq \{y \in \mathcal{X} : \mathbf{d}(x, y) \leq r\}$ are intervals in $(\mathcal{X}, \leq_{\mathbf{d}})$. In fact, this property uniquely characterizes linear Robinson dissimilarities. To prove this converse, the appropriate structure to analyze is the hypergraph $\mathcal{H}_{\mathbf{d}}$ with vertex set \mathcal{X} and hyperedge set $\mathbf{B}_{\mathbf{d}} \triangleq \{B_r^{\mathbf{d}}(x) : x \in \mathcal{X}, r > 0\}$. This hypergraph is called an *interval hypergraph* if every hyperedge is an interval [14].

Proposition 3.2 (see [19]). Let \mathbf{d} be a dissimilarity on \mathcal{X} . The following are equivalent:

1. \mathbf{d} is linear Robinson.
2. The hypergraph $\mathcal{H}_{\mathbf{d}}$ is an interval hypergraph.

It is easy to see that orderings consistent with a given linear Robinson dissimilarity are never unique. This follows from the *natural symmetries* of Robinson dissimilarities. Let $\leq_{\mathbf{d}}$ be consistent with respect to \mathbf{d} . Its *reversal* $\leq'_{\mathbf{d}}$ is the linear order defined by $x \leq'_{\mathbf{d}} y$ if and only if $y \leq_{\mathbf{d}} x$. It is clear that \mathbf{d} is linear Robinson with respect to $\leq_{\mathbf{d}}$ if and only if it is so with respect to $\leq'_{\mathbf{d}}$. Hence, the reversal of an ordering consistent with a dissimilarity is always consistent with the dissimilarity. Furthermore, there could be other consistent orderings in the case the dissimilarity is not strict, as we will see in [section 4](#).

3.1.2. Circular Robinson dissimilarities. Circular Robinson dissimilarities arise naturally when we allow for cyclic orders.

Definition 3.3 (the circular Robinson property). A dissimilarity \mathbf{d} on \mathcal{X} is circular Robinson if there exists a cyclic order $\mathcal{C}_{\mathbf{d}}$ such that

$$\forall \text{ cyclically ordered } w, x, y, z \in \mathcal{X} : \mathbf{d}(y, w) \geq \min\{\mathbf{d}(y, x), \mathbf{d}(y, z)\}.$$

We say it is strict circular Robinson if all the inequalities are strict. We say $\mathcal{C}_{\mathbf{d}}$ is consistent with \mathbf{d} and that \mathbf{d} is linear Robinson with respect to $\mathcal{C}_{\mathbf{d}}$.

Circular Robinson dissimilarities preserve the arcs of any compatible order, i.e., sets of the form $\{x \in \mathcal{X} : (m, x, M) \in \mathcal{C}_{\mathbf{d}}\}$ for $m, M \in \mathcal{X}$ called the borders of the arc. Arcs are the natural analogues of intervals for a cyclic order. Consequently, we say $\mathcal{H}_{\mathbf{d}}$ is an arc hypergraph if all its hyperedges are arcs. The analogue of Proposition 3.2 for a cyclic order is the following.

Proposition 3.4 (see [5, Proposition 5]). *Let \mathbf{d} be a dissimilarity. The following are equivalent:*

1. \mathbf{d} is circular Robinson.
2. The hypergraph $\mathcal{H}_{\mathbf{d}}$ is an arc hypergraph.

Similarly to the linear case, consistent orderings in the circular case are never unique. Let $\mathcal{C}_{\mathbf{d}}$ be consistent with respect to \mathbf{d} . In this case, its reversal $\mathcal{C}'_{\mathbf{d}}$ is the cyclic order such that $(x, y, z) \in \mathcal{C}'_{\mathbf{d}}$ if and only if $(z, y, x) \in \mathcal{C}_{\mathbf{d}}$. By definition, \mathbf{d} is circular Robinson with respect to $\mathcal{C}_{\mathbf{d}}$ if and only if it is so with respect to $\mathcal{C}'_{\mathbf{d}}$.

3.2. Robinson matrices. Let \mathbf{d} be a dissimilarity on \mathcal{X} . To any enumeration $x : [n] \rightarrow \mathcal{X}$ we can associate the $n \times n$ dissimilarity matrix D with entries $D(i, j) := \mathbf{d}(x_i, x_j)$. It is always nonnegative, symmetric, and with zero-diagonal. However, some enumerations will endow D with additional properties. This leads us to the concept of Robinson matrices.

3.2.1. Linear Robinson matrices. If \mathbf{d} is consistent with respect to $\leq_{\mathbf{d}}$ there exists an enumeration of \mathcal{X} such that for $i, j \in [n]$ we have $i \leq j$ if and only if $x_i \leq_{\mathbf{d}} x_j$. In this case, it follows that D induces a linear Robinson dissimilarity on $[n]$.

Definition 3.5 (linear Robinson matrix). *A dissimilarity matrix D is linear Robinson if*

$$(3.2) \quad \forall \text{ linearly ordered } i, j, k \in [n] : D(i, k) \geq \max\{D(j, i), D(j, k)\}.$$

It is strictly linear Robinson if all the inequalities are strict.

This implies D is consistent with the standard order on $[n]$, and Proposition 3.2 holds for D when $\mathcal{X} = [n]$.

When the dimension is understood from context, the set of linear and strictly linear Robinson dissimilarity matrices will be denoted \mathcal{L}_R and \mathcal{L}_R^* , respectively. Considering each one of these sets as \mathcal{M} leads to linear seriation and strict linear seriation, respectively.

Note that linear Robinson matrices inherit the symmetries from the dissimilarity. In fact, it can be verified that D is linear Robinson if and only if $\Pi_r D \Pi_r^T$ is linear Robinson. Since $\text{Dih}_1 \cong \langle \pi_r \rangle$, linear Robinson matrices are invariant under the action of Dih_1 by conjugation.

3.2.2. Circular Robinson matrices. For the circular case, we endow the set $[n]$ with the standard cyclic order \mathcal{C}_n

$$(3.3) \quad (i, j, k) \in \mathcal{C}_n \iff (i < j < k) \vee (j < k < i) \vee (k < i < j).$$

We still denote the standard linear order in $[n]$ as \leq .

If \mathbf{d} is consistent with respect to $\mathcal{C}_{\mathbf{d}}$ there exists an enumeration of \mathcal{X} such that $(i, j, k) \in \mathcal{C}_n$ if and only if $(x_i, x_j, x_k) \in \mathcal{C}_{\mathbf{d}}$. Similarly to the linear case, this implies D induces a circular Robinson dissimilarity on $[n]$.

Definition 3.6 (circular Robinson matrix). A dissimilarity matrix D is circular Robinson if

$$\forall \text{ cyclically ordered } i, j, k, \ell \in [n] : D(i, k) \geq \min\{D(j, k), D(k, \ell)\}.$$

Therefore, D is consistent with respect to \mathcal{C}_n , and [Proposition 3.4](#) holds for D when $\mathcal{X} = [n]$. When the dimension is understood from context, the set of circular and strictly circular Robinson matrices is denoted by \mathcal{C}_R and \mathcal{C}_R^* , respectively. Considering each one of these sets leads to the *circular seriation problem* and the *strict circular seriation*, respectively. Comparing [Definition 3.5](#) and [Definition 3.6](#), it is apparent that every linear Robinson matrix is also circular Robinson. In this sense, the notion of circular Robinson extends that of linear Robinson. The difference between linear and circular Robinson matrices is illustrated in [Figure 1](#).

We provide an alternative definition for circular Robinson matrices that will be useful in what follows. Let $f : [n] \rightarrow \mathbb{R}$. A *mode* is any $m \in [n]$ such that

$$\forall i, j \in [n] : (i \leq j \leq m \text{ or } m \leq j \leq i) \Rightarrow f_i \leq f_j \leq f_m.$$

We say f is *unimodal* if it has a mode. We say f is *strictly unimodal* if it has at most two distinct, consecutive modes $m_1 \leq m_2$ with $f_{m_1} = f_{m_2}$ and

$$\forall i, j \in [n] : (i < j < m_1 \Rightarrow f_i < f_j < f_{m_1}) \text{ and } (i > j > m_2 \Rightarrow f_i < f_j < f_{m_2}).$$

Notice that when $m_1 < m_2$ there are two modes, whereas when $m_1 = m_2$ there is only one mode. Our definition above allows us to treat both cases simultaneously. From the definition it is clear that every subsequence of a strictly unimodal sequence is also strictly unimodal. The proof of the following result is deferred to [Appendix A.1](#).

Proposition 3.7. Let D be a dissimilarity matrix. The following are equivalent:

1. D is circular Robinson (resp., strict circular Robinson).
2. For any $i \in [n]$ the function $j \rightarrow D(i, i+j \bmod n)$ is unimodal (resp., strict unimodal).

This property is naturally invariant under cyclic permutations.

Proposition 3.8. A dissimilarity matrix D is circular Robinson if and only if $\Pi_r D \Pi_r^T$ and $\Pi_s D \Pi_s^T$ are circular Robinson matrices.

Proof. First, notice that the (i, j) entry of $\Pi_s D \Pi_s^T$ and $\Pi_r D \Pi_r^T$ are $D(i+1 \bmod n, j+1 \bmod n)$ and $D(n-1-i, n-1-j)$, respectively. Noticing that $\{D(i \bmod n, i+j \bmod n)\}_{j=0}^{n-1}$ is unimodal for all i , we have that $\{D(i+1 \bmod n, i+j+1 \bmod n)\}_{j=0}^{n-1}$ and $\{D(n-1-i \bmod n, n-1-i+j \bmod n)\}_{j=0}^{n-1}$ are unimodal. Therefore $\Pi_r D \Pi_r^T$ and $\Pi_s D \Pi_s^T$ are also circular Robinson. ■

Since $\text{Dih}_n \cong \langle \pi_r, \pi_s \rangle$ it follows that circular Robinson matrices are invariant under the action of Dih_n by conjugation. This invariance is particular to the definition and should not be taken for granted. Other definitions proposed in the literature, e.g., [\[26\]](#), do not enjoy this property. We believe that cyclic invariance makes the definition arguably more natural.

3.3. Robinson orderings. The seriation problem does not assume we observe a linear or circular Robinson dissimilarity matrix, but instead its image under conjugation by an unknown permutation matrix. In other words, we observe matrices in $\text{pre-}\mathcal{L}_R$ and $\text{pre-}\mathcal{C}_R$. We call such matrices *Robinsonian matrices*.

Given a Robinsonian matrix and an algorithm for the corresponding seriation problem, the set of solutions may not be a singleton. In fact, the symmetries of linear and circular Robinson dissimilarity matrices ensure they will never be a singleton. We call *Robinson orderings* all the orderings represented by the elements in the set of solutions.

Although there will never be a unique Robinson ordering, we can at least distinguish which ones are due to the natural symmetries of the problem. Therefore, for the linear seriation problem we call solutions in the same orbit under the action Dih_1 the *trivial solutions*, whereas we call those in different orbits *nontrivial solutions*. The same criteria apply for the circular seriation problem when the action of Dih_n is considered instead.

4. The consecutive ones problem and PQ-trees. Robinson matrices turn out to be natural to formulate the seriation problem. We now review the connection between this problem and the *consecutive ones problem*. This connection yields polynomial time algorithms for solving the seriation problem and allows us to introduce PQ-trees, which will be extensively used in section 5.

4.1. The consecutive ones problem. The linear seriation problem is deeply connected to a combinatorial problem known as the *consecutive ones (C1) problem*. To introduce this problem, consider an $m \times n$ binary matrix M . The C1 problem is to find a permutation Π such that the entries of $M\Pi$ equal to one appear consecutively along rows. We say M has the *consecutive ones (C1) property* if the C1 problem has a solution for M . An example of such matrix can be found in Figure 2(a). The first linear time algorithm for the C1 problem was introduced by Booth and Lueker in [3]. If f is the number of ones in M , then their result states the C1 problem can be decided in $\mathcal{O}(m + n + f)$ time.

An extension to this problem is the *circular ones (Cr1) problem*. The Cr1 problem is to find a permutation Π such that the entries of $M\Pi$ equal to one appear consecutively modulo n along rows. We say M has the *circular ones (Cr1) property* if the Cr1 problem has a solution for M [29]. This problem can also be solved efficiently as it can be reduced to the C1 problem. Let \overline{M} be the matrix such that every row with a 1 on its first entry is complemented. Then M satisfies the Cr1 property if and only if \overline{M} satisfies the C1 property [29, Theorem 1]. Therefore, by forming the complement, the Cr1 problem can be decided in polynomial time.

Both problems are connected to the seriation problem through the ball hypergraph \mathcal{H}_D , introduced in subsection 3.1. In fact, interval and arc hypergraphs are precisely those for which their incidence matrices, respectively, satisfy the C1 and Cr1 properties [14]. This suggests how to efficiently solve the seriation problem for Robinson matrices.

Theorem 4.1 (see [6, 19]). *The linear and circular seriation problem can be reduced in polynomial time and space to deciding, respectively, the C1 and Cr1 problem. Robinson matrices can be recognized in $\mathcal{O}(n^3)$ time and with $\mathcal{O}(n^3)$ space.*

The bounds above follow from the worst case in which \mathcal{H}_D has $\mathcal{O}(n^2)$ different hyperedges. In this case, for each of the n possible centers and each row $i \in [n]$ the matrix can take $\mathcal{O}(n)$ possible values. In this case, the incidence matrix has $\mathcal{O}(n^3)$ entries.

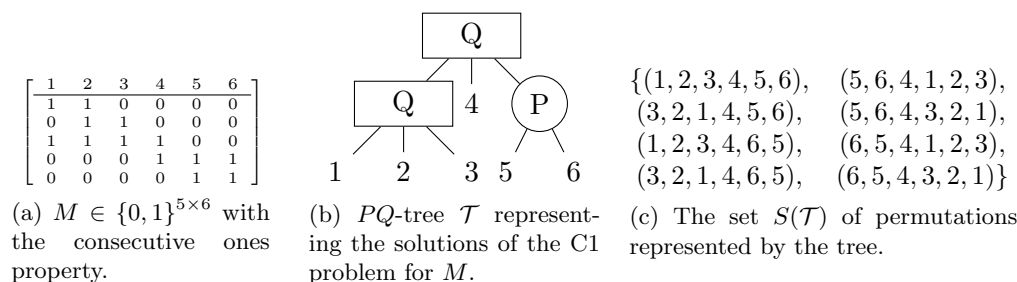


Figure 2. A PQ -tree of all solutions to the C1 problem for a $\{0,1\}$ -matrix.

4.2. PQ -trees. The algorithmic structure underlying the algorithm to solve the C1 problem is the PQ -tree. A PQ -tree \mathcal{T} on a set \mathcal{X} is a rooted tree with two types of internal nodes denoted by P , represented as circles, and Q , represented as rectangles, and where the leaves represent the elements in \mathcal{X} . The type of node represents admissible permutations on \mathcal{X} : children of a P -node can be permuted arbitrarily, whereas children of a Q -node can only be reversed. Figure 2(b) shows an example of a PQ -tree.

PQ -trees are related to the C1 problem as follows. Let Y_i be the indices of the columns of M such that its i th entry is equal to one. Then $\mathbf{Y} = \{Y_i\}_{i \geq 0}$ is a collection of subsets of $[n]$. The C1 problem can be solved if we can permute the elements of $[n]$ so that every Y_i becomes an interval. The algorithm starts with a single set $\mathbf{Y}_1 = \{Y_{i_1}\}$ and determines the set of *admissible permutations* such that Y_{i_1} becomes an interval. These can be represented by a PQ -tree \mathcal{T}_1 (see [3] and [4]). The algorithm proceeds by adding a Y_{i_2} to form $\mathbf{Y}_2 = \{Y_{i_1}, Y_{i_2}\}$ and update the PQ -tree accordingly. The main contribution of [4] is an algorithm for updating \mathcal{T}_k in a way that given any subset $Y_k \subseteq [n]$, the set of permutations represented by the updated tree \mathcal{T}_{k+1} is precisely the set of admissible permutations of $\mathbf{Y}_{k+1} \cup \{Y_k\}$. This is done in time linear in the size of Y_k . The algorithm finishes when \mathbf{Y} is attained.

As an example, by considering all rows of the binary matrix in Figure 2(a), the resulting PQ -tree at the final step would be the one in Figure 2(b), and the solution set would be the one in Figure 2(c).

5. Optimal algorithm for strict circular seriation. In this section, we present an optimal algorithm for circular seriation in the strict Robinson case. We note in passing that our algorithm works as well for the strict linear case, but we omit this variant. Our algorithm runs in $\mathcal{O}(n^2)$ time and space, which is obviously optimal, since it is the time required to read the input and the space required to provide a strict Robinson dissimilarity.² The core algorithm relies on two main ideas: merging nearest neighbors and discarding forbidden arc reversals. We recursively merge nearest neighbors, using the fact that nearest neighbors are guaranteed to be consecutive elements in a strict Robinsonian ordering. Exploiting this fact we can obtain chains of consecutive elements which are stored in Q -nodes of PQ -trees. The most delicate part of the algorithm consists in efficiently deciding whether each Q -node can

²Given the strict Robinson property, it is clear that the underlying matrix is dense, and therefore $\mathcal{O}(n^2)$ memory is required to even provide the input.

be *uniquely oriented*. In that case, such Q -node can be deleted and its children merged to the parent Q -node. We will indistinctly refer to this operation as the Q -node being *fixed* or *oriented*. The process of building chains of consecutive elements and deciding their orientation can be done in several ways. The advantage of our algorithm is that the total number of comparisons to decide each orientation is bounded by $\mathcal{O}(n)$, which leads to a running time $\mathcal{O}(n^2)$.

We provide preliminary results in [subsection 5.1](#) and [subsection 5.2](#) about the arc structure of the nearest-neighbor graph and the treatment of arc reversals. Then, in [subsection 5.3](#) we develop our algorithm, whose correctness and optimality are proved in [subsection 5.4](#).

5.1. Preliminaries part I: Nearest neighbors graph in strict Robinson dissimilarities.

One of our algorithmic building blocks is based on the idea that in the case of *strict dissimilarities*, a pair of nearest neighbors must lie consecutively in *any* Robinson ordering. In this subsection, we prove this fact and use it to justify the first step of our algorithm, which is based on merging nearest neighbors to reduce the size of the instance.

Given $D \in \text{pre-}\mathcal{C}_R$, a collection of subsets $\mathcal{P} = \{\mathcal{I}_i\}_{i \geq 0}$ of \mathcal{X} is said to be an arc partition if \mathcal{P} is a partition of \mathcal{X} and every set \mathcal{I}_i is an arc of consecutive elements in any Robinson ordering. The set of nearest neighbors of $x \in \mathcal{X}$ is defined as $\text{NN}(x) \triangleq \arg \min_{y \in \mathcal{X} \setminus \{x\}} \mathbf{d}(x, y)$. The nearest-neighbors graph is an undirected graph $G_{\text{NN}}(\mathcal{X}, \mathbf{d}) = (\mathcal{X}, \mathcal{E})$ such that $\{x, y\} \in \mathcal{E}$ if and only if $x \in \text{NN}(y)$ or $y \in \text{NN}(x)$. An essential condition of strict Robinson dissimilarities is what we called the *nearest-neighbor condition*, which implies that the connected components of the nearest-neighbors graph correspond to arcs of consecutive elements, and since connected components form a partition, such a collection corresponds to an arc partition.

Definition 5.1 (nearest-neighbor condition). *A dissimilarity matrix $D \in \mathbb{R}^{n \times n}$ is said to have the nearest-neighbor condition if it holds that $\text{NN}(i) \subseteq \{i - 1 \bmod n, i + 1 \bmod n\}$.*³

It is immediate to verify that *strict* circular Robinson dissimilarities satisfy the nearest-neighbor condition, which is not necessarily true in the nonstrict case.

We also recall from graph theory that given a graph $G = (\mathcal{X}, \mathcal{E})$ and a node $x \in \mathcal{X}$, the set of adjacent nodes to x is denoted as $\mathcal{N}_G(x) \triangleq \{y \in \mathcal{X} : \{x, y\} \in \mathcal{E}\}$. The function $x \mapsto \mathcal{N}_G(x)$ is called the neighborhood. The cycle graph $C_n = ([n], \mathcal{E})$ is the graph with edge set $\mathcal{E} = \{\{i, (i + 1) \bmod n\} : i \in [n]\}$. If a graph G is a subgraph of C_n , then it is clear that its connected components correspond to arcs of $([n], \mathcal{C}_n)$. A direct consequence of the nearest-neighbor condition is that the nearest-neighbors graphs of *strict* circular Robinson dissimilarity matrices correspond to subgraphs of C_n .

Our algorithm relies crucially on the fact that strict dissimilarities must respect nearest neighbors in any Robinson ordering. This is not necessarily true in the nonstrict case.

Lemma 5.2. *Let $D \in \text{pre-}\mathcal{C}_R^*$ and $i \in [n]$. Suppose that $j \in \text{NN}(i)$; then in any Robinson ordering σ , the elements i and j are consecutive.*

Proof. Let σ be any Robinson ordering. Let $j \in \text{NN}(i)$, and let $r \triangleq D(i, j)$. This implies that for any $k \in B(i, r) \setminus \{i\}$, $D(i, k) = D(i, j)$. Suppose by contradiction that there exist k_1, i, k_2 consecutive in σ , with $j \neq k_1, k_2$. Since in any Robinson ordering balls are arcs, this

³Given some enumeration $\#$ and $i \in [n]$, when we write $\text{NN}(i)$, we refer to the set $\#(\text{NN}(x_i))$.

implies that either $k_1 \in B(i, r)$ or $k_2 \in B(i, r)$. Any of the two cases is a contradiction with the nearest-neighbor condition, proving the result. ■

Since nearest-neighbors must be consecutive, we get that the connected components of the nearest-neighbors graph of a *strict* circular Robinson dissimilarity correspond to arcs of any Robinson ordering. Hence, the set of connected components constitutes an arc partition. The fact that this graph is a subgraph of the cycle graph makes computationally efficient finding the order intrinsic to each component, and the task is divided in two steps:

1. Find all degree 1 nodes. These correspond to the borders of the components.
2. Perform depth-first search (DFS) ([Algorithm B.2](#)) starting at each nonvisited degree 1 node. The order of visits will follow the Robinson ordering (or backwards).

If there are no degree one nodes, then⁴ $G_{\text{NN}} \cong C_n$ and therefore we can start at any node. For an algorithmic implementation, tuples can be used to represent the local fragments of Robinson orderings (Q -nodes). A tuple is an ordered set $\alpha = (a_0, a_1, a_2, \dots, a_{k-1})$. We write $\alpha(i)$ to denote a_i , the i th element of α . Each connected component will be stored in a tuple α , where $\alpha(j)$ is the j th element visited by performing a DFS. The procedure is summarized in the procedure [Arc Partition](#) ([Algorithm B.1](#)), whose correctness is stated in the following proposition (the proof of the next result is omitted for brevity).

Proposition 5.3. *Given a matrix $D \in \text{pre-}\mathcal{C}_R^*$, by performing [Arc Partition](#) with input $([n], D)$ the resulting tuples follow an arc ordering for every Robinson ordering.*

5.2. Preliminaries part II: Orienting arcs. The previous section tell us that nearest neighbors must be consecutive in the strict Robinson case. By exploiting this idea we can obtain ordered sequences of elements stored in Q -nodes of a PQ -tree. Notice, however, that Q -nodes are allowed to be reversed, which at this point of the algorithm is not guaranteed to lead to Robinson orderings. If this is not the case, the inconsistent ordering must be discarded, which corresponds to removing the Q -node and merging its children directly to the parent Q -node. We call this process *orientation*. In this section we provide computationally efficient conditions to determine when a Q -node must be oriented. Each Q -node α in a tree \mathcal{T} can be associated with an arc \mathcal{I}_α in \mathcal{X} : the arc of all leaves in \mathcal{X} which are descendants of α . Reversing α corresponds to reversing \mathcal{I}_α . The first relevant concept to determine when it is possible to reverse each arc is the strictly overlapping condition, which has been studied, for instance, in [23] and in [14]. An example of the property can be seen in [Figure 3\(a\)](#).

Definition 5.4. *Two arcs \mathcal{I} and \mathcal{J} are said to strictly overlap, denoted by $\mathcal{I} \not\propto^* \mathcal{J}$, if*

1. $\mathcal{I} \not\subset \mathcal{J}$, 2. $\mathcal{J} \not\subset \mathcal{I}$, 3. $\mathcal{I}^c \not\subset \mathcal{J}$, and 4. $\mathcal{J} \not\subset \mathcal{I}^c$.

Observation 5.5. The relation $\not\propto^*$ is symmetric and equivalent to

1. $\mathcal{I} \cap \mathcal{J}^c \neq \emptyset$, 2. $\mathcal{J} \cap \mathcal{I}^c \neq \emptyset$, 3. $\mathcal{I}^c \cap \mathcal{J}^c \neq \emptyset$, and 4. $\mathcal{I} \cap \mathcal{J} \neq \emptyset$.

Lemma 5.6. *Let \mathcal{I} and \mathcal{J} be two arcs. Let a, b and a', b' be the borders of \mathcal{I} and \mathcal{I}^c , respectively, where a and a' (b and b') are consecutive in the cyclic order. Then $\mathcal{I} \not\propto^* \mathcal{J}$ if and only if one of the following conditions holds:*

⁴We use the symbol \cong to denote either graph or group isomorphism.

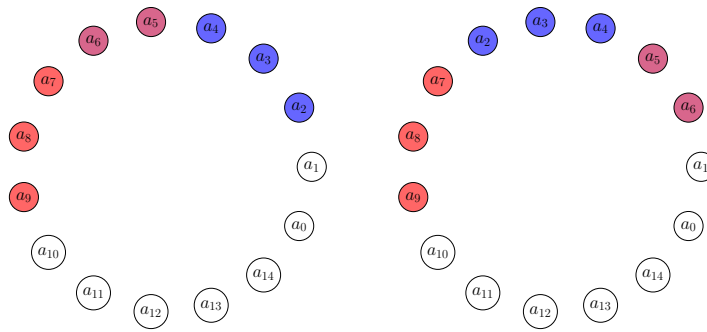


Figure 3. In the left, two strictly overlapping arcs $\mathcal{I} = \{a_i\}_{i=2}^6$ (blue and purple) and $\mathcal{J} = \{a_i\}_{i=5}^9$ (red and purple). The intersection $\mathcal{I} \cap \mathcal{J} = \{a_5, a_6\}$ in purple. In the right, we have the ordering after the action of the permutation σ that reverses the elements of \mathcal{I} .

$$(i) \{a, a'\} \subset \mathcal{J} \text{ and } \{b, b'\} \subset \mathcal{J}^c \quad \text{or} \quad (ii) \{b, b'\} \subset \mathcal{J} \text{ and } \{a, a'\} \subset \mathcal{J}^c.$$

Proof. We first prove (\Leftarrow) . Suppose (i) holds (the other case follows analogously). Then since $a \in \mathcal{I}$ and $a' \in \mathcal{I}^c$ we get conditions 2 and 4 of [Observation 5.5](#). Now, since $b \in \mathcal{I}$ and $b' \in \mathcal{I}^c$ we get conditions 1 and 3 of [Observation 5.5](#).

Next we prove (\Rightarrow) . First we notice that there are at least two elements in \mathcal{I} and two elements in \mathcal{I}^c (otherwise containing a single element of these arcs would imply containing the whole set, contradicting one of the conditions in [Definition 5.4](#)). Hence, the elements a, b, a' , and b' exist and are distinct. Suppose $a \in \mathcal{J}$ (the case $a \in \mathcal{J}^c$ is analogous), and let $z \in \mathcal{J} \setminus \mathcal{I}$ (exists by hypothesis). Since \mathcal{J} is an arc, it must contain one of the two paths connecting a and z . Since it does not contain the whole \mathcal{I} , it must be the path that covers a' ; therefore, $\{a, a'\} \subset \mathcal{J}$ and $b \notin \mathcal{J}$. On the other hand, since it does not contain the whole \mathcal{I}^c , $b' \notin \mathcal{J}$. It follows that $\{b, b'\} \subset \mathcal{J}^c$. ■

Given an arc $\mathcal{I} = \{a_0, \dots, a_{k-1}\}$ (where elements are indexed following the cyclic order), we define the *permutation that reverses* \mathcal{I} as the permutation σ such that $\sigma(a_j) = a_{k-j-1}$ for $j \in \{0, \dots, k-1\}$ and $\sigma(x) = x$ if $x \notin \mathcal{I}$.

Lemma 5.7. *Let \mathcal{I} and \mathcal{J} be two arcs, and let σ be the permutation that reverses the elements of \mathcal{I} . Then $\mathcal{I} \check{\simeq}^* \mathcal{J}$ if and only if the permutation of \mathcal{J} by σ is not an arc.*

Proof. We first prove (\Leftarrow) . By contraposition, assume any of the conditions in [Definition 5.4](#) do not hold; then it is easy to see that $\sigma(\mathcal{J}) = \mathcal{J}$, which is an arc. Now we prove (\Rightarrow) . If $\mathcal{I} \check{\simeq}^* \mathcal{J}$, then at least one of the conditions of [Lemma 5.6](#) hold. Since $\sigma(a) = b$, $\sigma(b) = a$, $\sigma(a') = a'$, and $\sigma(b') = b'$, then $\sigma(\mathcal{J})$ is not connected and thus it is not an arc. ■

As an example, consider the two strictly overlapping arcs \mathcal{I} and \mathcal{J} in [Figure 3](#). By reversing \mathcal{I} , the arc \mathcal{J} gets ripped apart into two disconnected pieces: namely its red nodes and its purple nodes. Recall from [Proposition 3.4](#) that a dissimilarity matrix is circular Robinson if and only if each ball \mathcal{J} is an arc. Therefore, any arc \mathcal{I} cannot be arbitrarily reversed to produce a new Robinson ordering if and only if there is some ball that strictly overlaps with \mathcal{I} . In terms of PQ -trees, a necessary and sufficient condition for a Q -node α to be orientable is the existence of some $z \in \mathcal{X}$ and $r > 0$ such that the ball $B_r(z)$ strictly overlaps with \mathcal{I}_α . In

such case, one of the two orientations of the node is not compatible with a Robinson ordering since in one of these orientations the ball gets disconnected. By [Lemma 5.6](#), to determine the orientation of the arc \mathcal{I} one could equivalently check whether there exists $z \in \mathcal{X}$ and $r > 0$ such that

$$[\{a, a'\} \subset B_r(z) \wedge \{b, b'\} \subset B_r(z)^c] \quad \vee \quad [\{b, b'\} \subset B_r(z) \wedge \{a, a'\} \subset B_r(z)^c].$$

If none of these conditions hold, then we say the arc is *not orientable* which means that the Q -node in the tree must be preserved. Notice that this requires knowing that a, b (a', b') are the borders of \mathcal{I} (resp., \mathcal{I}^c) in advance. **Border Candidates Orientation** ([Algorithm 5.1](#)) is an efficient way for orienting the arc \mathcal{I} with respect to the dissimilarity \mathbf{d} when we have *border candidates* but the actual borders within the candidates are unknown.

Definition 5.8 (border candidates of an arc). A 4-tuple of sets $(\mathcal{A}', \mathcal{A}, \mathcal{B}, \mathcal{B}')$ are said to be *border candidates of the arc \mathcal{I}* if the following properties hold:

1. $\mathcal{A}, \mathcal{B} \subset \mathcal{I}$ and $\mathcal{A}', \mathcal{B}' \subset \mathcal{I}^c$.
2. The sets are pairwise disjoint.
3. If a, b are the borders of \mathcal{I} and a', b' are the borders of \mathcal{I}^c , then $a \in \mathcal{A}, b \in \mathcal{B}, a' \in \mathcal{A}'$, and $b' \in \mathcal{B}'$.
4. Either $(\mathcal{A}', \mathcal{A}, \mathcal{B}, \mathcal{B}')$ or $(\mathcal{A}', \mathcal{B}, \mathcal{A}, \mathcal{B}')$ is cyclically ordered.⁵

The next result provides correctness for **Border Candidates Orientation** ([Algorithm 5.1](#)) (see its proof in [Appendix A.2](#)).

Lemma 5.9. Let $\mathcal{I} \subset \mathcal{X}$ be an arc in any Robinson ordering. Suppose a, b are the borders of \mathcal{I} and a', b' are the respective borders of \mathcal{I}^c . Additionally suppose that $(\mathcal{A}', \mathcal{A}, \mathcal{B}, \mathcal{B}')$ are border candidates for \mathcal{I} . Then, for every $z \in \mathcal{X}$, the following statements are equivalent:

1. Both $\{a, a'\} \subset B_r(z)$ and $\{b, b'\} \subset B_r(z)^c$ hold for some $r > 0$.
2. There exist $(x', x, y, y') \in \mathcal{A}' \times \mathcal{A} \times \mathcal{B} \times \mathcal{B}'$ such that $\max\{f_z(x), f_z(x')\} < \min\{f_z(y), f_z(y')\}$.
3. It holds that $\max\{\min f_z(\mathcal{A}), \min f_z(\mathcal{A}')\} < \min\{\max f_z(\mathcal{B}), \max f_z(\mathcal{B}')\}$.

Above $f_z(\cdot) \triangleq \mathbf{d}(z, \cdot)$ and given any $U \subset \mathcal{X}$, $\min f_z(U) \triangleq \min_{y \in U} f_z(y)$ (similar for \max).

Corollary 5.10. Let $\mathcal{I} \subset \mathcal{X}$ be an arc in any Robinson ordering, and suppose the sets $(\mathcal{A}', \mathcal{A}, \mathcal{B}, \mathcal{B}')$ are border candidates for the arc \mathcal{I} . Then, [Algorithm 5.1](#) correctly determines if \mathcal{I} must be fixed or reversed or if it is not orientable.

Observation 5.11. The time complexity of [Algorithm 5.1](#) with input $(\mathcal{A}', \mathcal{A}, \mathcal{B}, \mathcal{B}')$ is $\mathcal{O}(|\mathcal{X}| \cdot \max\{|\mathcal{A}'|, |\mathcal{A}|, |\mathcal{B}|, |\mathcal{B}'|\})$; thus it is an efficient way of orienting a Q -node α whenever the set of border candidates for \mathcal{I}_α is not too big.

5.3. The recursive seriation algorithm. We devote the next subsection to describing our main algorithm: **Recursive Seriation** ([Algorithm 5.2](#)). This algorithm starts from the singleton elements of \mathcal{X} and proceeds by recursively detecting nearest neighbors among previously computed arcs, merging them, and deciding whether the merged sets must be oriented. An important ingredient of our algorithm is the use of Q -trees: this is a useful data structure to

⁵Formally, the ordered collection is a consistent cyclic quasi-order; see [Definition 5.12](#).

Algorithm 5.1 Border Candidates Orientation

```

1: Input: A sequence of sets  $(\mathcal{A}', \mathcal{A}, \mathcal{B}, \mathcal{B}')$ 
2: Let  $f_z(x) \triangleq \mathbf{d}(z, x)$  for every  $z \in \mathcal{X}$ 
3: Let  $O_i : \mathcal{X} \rightarrow \{True, False\}$  for  $i = 1, 2, 3, 4$  be defined by
4:  $O_1(z) \triangleq \max \{ \min f_z(\mathcal{A}), \min f_z(\mathcal{A}') \} < \min \{ \max f_z(\mathcal{B}), \max f_z(\mathcal{B}') \}$ 
5:  $O_2(z) \triangleq \max \{ \min f_z(\mathcal{B}), \min f_z(\mathcal{B}') \} < \min \{ \max f_z(\mathcal{A}), \max f_z(\mathcal{A}') \}$ 
6:  $O_3(z) \triangleq \max \{ \min f_z(\mathcal{A}), \min f_z(\mathcal{B}') \} < \min \{ \max f_z(\mathcal{A}'), \max f_z(\mathcal{B}) \}$ 
7:  $O_4(z) \triangleq \max \{ \min f_z(\mathcal{B}), \min f_z(\mathcal{A}') \} < \min \{ \max f_z(\mathcal{B}'), \max f_z(\mathcal{A}) \}$ 
8: for  $z \in \mathcal{X}$  do
9:   if  $O_1(z) \vee O_2(z)$  then
10:    return 'correct'
11:   else if  $O_3(z) \vee O_4(z)$  then
12:    return 'reverse'
13:   end if
14: end for
15: return 'not orientable'
16: Output: A string determining the orientation of the input

```

merge arcs, as well as fixing the orientations when they are detected. In order to run in total quadratic time, great care is needed regarding the number of comparisons used to decide the orientations, for which we find it useful to compute *border candidates* for Q -trees.

For the sake of readability, we dissect the main algorithm in terms of various subroutines, which are presented and analyzed separately. Hence, we structure our presentation as follows. In [subsection 5.3.1](#) we specify the initialization of `Recursive Seriation`. In [subsection 5.3.2](#), we describe the key operation of computing the border candidates of a Q -tree, summarized in the procedure `Border Candidates` ([Algorithm 5.3](#)). Next, in [subsection 5.3.3](#) we describe the operation of computing dissimilarities among Q -trees and how this leads to the recursion behind `Recursive Seriation`. In [subsection 5.3.4](#) we study the process of deciding the orientation among consecutive children of a given Q -node, including a pseudocode of this procedure `Consecutive Orientation` ([Algorithm 5.4](#)). Next, in [subsection 5.3.5](#) we use the previous subroutine and complement it with the additional steps required to completely orient the internal nodes of a Q -tree; the associated subroutines here are in the procedures `Complete Internal Orientation` ([Algorithm 5.5](#)) and `Final Orientation` ([Algorithm 5.6](#)). In [subsection 5.3.6](#) we provide a method to decide the orientation between trees that have been detected to be nearest neighbors; the useful subroutine here is the `External Orientation` ([Algorithm 5.7](#)).

5.3.1. Initialization. `Recursive Seriation` receives as input a family \mathbf{T} of Q -trees (which are PQ -trees composed solely by Q -nodes) with leaves corresponding to elements of \mathcal{X} . Given an instance $(\mathcal{X}, \mathbf{d})$, we initialize the algorithm with $(\mathcal{X}, \mathbf{d}, \mathbf{T})$, where $\mathbf{T} = \{x : x \in \mathcal{X}\}$. For each $\mathcal{T} \in \mathbf{T}$ we write as $\partial\mathcal{T}$ the set of leaves of \mathcal{T} . Hence, in the initial case we get $\partial x \triangleq \{x\}$.

5.3.2. Computing border candidates. We endow each $\mathcal{T} \in \mathbf{T}$ with a set $\mathcal{B}(\mathcal{T})$ of *border candidates*,⁶ which are all leaves of \mathcal{T} that appear in the extreme left or right under some

⁶We emphasize the distinction of the border candidates *of a tree*, which we are about to introduce, and the border candidates *of an arc*, introduced in [Definition 5.8](#).

Algorithm 5.2 Recursive Seriation

```

1: Input:  $(\mathcal{X}, \mathbf{d}, \mathbf{T})$ :  $\mathbf{d}$  dissimilarity over  $\mathcal{X}$ ;  $\mathbf{T}$  family of  $Q$ -trees with leaves given by elements of  $\mathcal{X}$ 
2: for  $\mathcal{T} \in \mathbf{T}$  do
3:   for  $\mathcal{T}' \in \mathbf{T}$  do
4:      $\mathbf{d}^{\min}(\mathcal{T}, \mathcal{T}')$ ,  $\mathbf{d}^{\arg \min}(\mathcal{T}, \mathcal{T}') = \text{Tree Dissimilarity}(\mathcal{T}, \mathcal{T}', \mathbf{d})$ 
5:      $\text{External Orientation}(\mathcal{T}, \mathcal{T}', \mathbf{d}^{\arg \min}(\mathcal{T}, \mathcal{T}'))$ 
6:   end for
7: end for
8:  $\mathbf{T}' = \text{Arc Partition}(\mathbf{d}^{\min}, \mathbf{T})$ 
9: if  $|\mathbf{T}'| = 1$  then
10:   $\text{Final Orientation}(\mathcal{T})$  {where  $\mathbf{T}' = \{\mathcal{T}\}$ }
11:  return  $\mathcal{T}$ 
12: else
13:   for  $\mathcal{T} \in \mathbf{T}'$  do
14:      $\text{Complete Internal Orientation}(\mathcal{T})$ 
15:   end for
16:   return  $\text{Recursive Seriation}(\mathcal{X}, \mathbf{d}, \mathbf{T}')$ 
17: end if
18: Output: A  $Q$ -tree  $\mathcal{T}$  containing all Robinson orderings

```

configuration of the tree. Whenever $|\partial\mathcal{T}| \geq 2$, and conditionally on an given orientation of the root, the set of border candidates $\mathcal{B}(\mathcal{T})$ can be split in two: left and right. The set of left border candidates, denoted as $\mathcal{B}^L(\mathcal{T})$, are all elements in $\mathcal{B}(\mathcal{T})$ that appear in the extreme left under some configuration of the tree. The definition of $\mathcal{B}^R(\mathcal{T})$ is entirely analogous with appearing on the right. For instance, in the tree \mathcal{T} appearing in Figure 5(a), we have $\mathcal{B}^L(\mathcal{T}) = \{a_3, b_3, b_2, b_1\}$ and $\mathcal{B}^R(\mathcal{T}) = \{b_0\}$.

Let α denote the root of \mathcal{T} . Whenever $\text{depth}(\mathcal{T}) > 1$, where $\text{depth}(\mathcal{T})$ denotes the tree-depth of \mathcal{T} , \mathcal{T}^L (resp., \mathcal{T}^R) denotes the subtree of \mathcal{T} whose root is the first (resp., last) Q -node among the direct descendants of α . Notice that for the computation of the border candidates of \mathcal{T} it is convenient to consider the relations $\mathcal{B}^L(\mathcal{T}) = \mathcal{B}(\mathcal{T}^L)$ and $\mathcal{B}^R(\mathcal{T}) = \mathcal{B}(\mathcal{T}^R)$. Thus, to obtain $\mathcal{B}(\mathcal{T}) = \mathcal{B}(\mathcal{T}^L) \cup \mathcal{B}(\mathcal{T}^R)$ it suffices to recursively call for the border candidates of the subtrees \mathcal{T}^L and \mathcal{T}^R . **Border Candidates** (Algorithm 5.3) is a straightforward implementation of this idea that runs in time $\mathcal{O}(|\mathcal{B}(\mathcal{T})|)$.

Algorithm 5.3 Border Candidates

```

1: Input: A  $Q$ -tree  $\mathcal{T}$ 
2: if  $\text{depth}(\mathcal{T}) = 0$  then
3:   return  $\mathcal{T}$  {Return the single element from the tree}
4: else
5:   return  $\text{Border Candidates}(\mathcal{T}^L) \cup \text{Border Candidates}(\mathcal{T}^R)$ 
6: end if
7: Output: The set  $\mathcal{B}(\mathcal{T})$  of border candidates of  $\mathcal{T}$ 

```

5.3.3. Computing the minimum pairwise dissimilarity among trees. A second key step is to define an appropriate dissimilarity between trees. This allows us to solve the problem

recursively by decreasing the number of objects we need to sort in each iteration. In the initial case we have that this dissimilarity corresponds to \mathbf{d} , and since it is strict circular Robinson, `Arc Partition` (available in [Appendix B](#)) with input (\mathbf{T}, \mathbf{d}) returns an arc partition of \mathbf{T} stored in tuples. The elements in each tuple α must be consecutive. Therefore, for each α we build a new tree \mathcal{T}_α with a Q -node in the root whose i th child corresponds to $\alpha(i)$. The recursion works by repeating the process over the smaller family of trees $\mathbf{T}' = \{\mathcal{T}_\alpha\}$ until $|\mathbf{T}'| = 1$.

For this to work we need to define a dissimilarity \mathbf{d}' over \mathbf{T}' in a such way that sorting \mathbf{T}' with respect to \mathbf{d}' helps us in our goal of sorting \mathbf{T} with respect to \mathbf{d} . We now present a dissimilarity that does exactly that: given two trees $\mathcal{T}_1, \mathcal{T}_2$, consider the dissimilarity $\mathbf{d}^{\min}(\mathcal{T}_1, \mathcal{T}_2) \triangleq \min\{\mathbf{d}(x, y) : x \in \partial\mathcal{T}_1, y \in \partial\mathcal{T}_2\}$. Also let $\mathbf{d}^{\arg\min}(\mathcal{T}_1, \mathcal{T}_2)$ be the collection of all minimizers of this problem in $\partial\mathcal{T}_1 \times \partial\mathcal{T}_2$. The following lemma justifies this choice, as it ensures that by sorting \mathbf{T}' we obtain a quasi-order among the elements of our original set \mathbf{T} .

Definition 5.12 (quasi-order⁷). *Let $(\mathcal{X}, \mathcal{C})$ be a cyclically ordered set. An ordered partition $\{A_0, \dots, A_{m-1}\}$ is a (consistent) cyclic quasi-order if for all $(i, j, k) \in \mathcal{C}_m$, $x \in A_i$, $y \in A_j$, and $z \in A_k$ we have that $(x, y, z) \in \mathcal{C}$.*

Lemma 5.13. *Let $D \in \mathbb{R}^{n \times n}$ be a (strict) circular Robinson dissimilarity, and let $\{A_i\}_{i \in [m]}$ be a cyclic quasi-order in $[n]$. The matrix $D^{\min}(A_i, A_j) \triangleq \min\{D(k, l) : k \in A_i, l \in A_j\}$ is a (strict) circular Robinson dissimilarity.*

Since nearest neighbors must be consecutive, the family of Q -trees \mathbf{T} at each recursion is guaranteed to satisfy that $\{\partial\mathcal{T}\}_{\mathcal{T} \in \mathbf{T}}$ is an arc partition. Therefore, by the previous lemma we have that $D'(i, j) \triangleq \mathbf{d}^{\min}(\mathcal{T}_i, \mathcal{T}_j) \in \text{pre-}\mathcal{C}_R^*$ (pre- \mathcal{C}_R) whenever the original dissimilarity matrix D is in pre- \mathcal{C}_R^* (pre- \mathcal{C}_R) and a Robinson ordering for D' yields a quasi-order for D .

A naïve computation of \mathbf{d}^{\min} does not lead to a global $\mathcal{O}(n^2)$ time complexity. This is why our next goal is to find a workaround. The next result, which is a direct consequence of [Proposition A.1](#) in [Appendix A.1](#), implies that we can reduce this search by only considering border candidates of each tree. In particular, $\mathbf{d}^{\arg\min}(\mathcal{T}_1, \mathcal{T}_2) \subseteq \mathcal{B}(\mathcal{T}_1) \times \mathcal{B}(\mathcal{T}_2)$.

Lemma 5.14. *Suppose $D \in \mathcal{C}_R^*$, and let $\mathcal{I} = [a, b]$ be an arc in $([n], \mathcal{C}_n)$. Then for every $i \in \mathcal{I}^c$, all minimizers of $\min\{D(i, j) : j \in \mathcal{I}\}$ are contained in $\{a, b\}$.*

Observation 5.15. Let `Tree Dissimilarity` be the procedure which receives a pair $(\mathcal{T}_1, \mathcal{T}_2)$ of Q -trees and a dissimilarity \mathbf{d} to return $\mathbf{d}^{\min}(\mathcal{T}_1, \mathcal{T}_2)$ and $\mathbf{d}^{\arg\min}(\mathcal{T}_1, \mathcal{T}_2)$ by brute force comparisons among border candidates. It takes $\mathcal{O}(|\mathcal{B}(\mathcal{T}_1)| \cdot |\mathcal{B}(\mathcal{T}_2)|)$ operations and $\mathcal{O}(1)$ space, since by [Lemma 5.14](#), $|\mathbf{d}^{\arg\min}(\mathcal{T}_1, \mathcal{T}_2)| \leq 4$. We omit its pseudocode for brevity.

Now we introduce the main procedure required for orienting Q -nodes within the trees.

5.3.4. Consecutive Q -nodes orientation. Recall from [Corollary 5.10](#) that in order to orient the root of a Q -tree \mathcal{T}_2 , it suffices to find border candidates for the arc $\partial\mathcal{T}_2$. Let

⁷This extends the definition introduced in [\[6\]](#) for linear orders to cyclic orders.

$\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3$ be children of a Q -node α such that \mathcal{T}_2 succeeds \mathcal{T}_1 and precedes \mathcal{T}_3 ; then we decide their orientation with the Consecutive Orientation procedure (Algorithm 5.4).

Algorithm 5.4 Consecutive Orientation

- 1: **Input:** Three consecutive subtrees $(\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3)$ of a Q -node α
 - 2: Let $\mathcal{A}' \triangleq \mathcal{B}(\mathcal{T}_1)$, $\mathcal{B}' \triangleq \mathcal{B}(\mathcal{T}_3)$, $\mathcal{A} \triangleq \mathcal{B}^L(\mathcal{T}_2)$ and $\mathcal{B} \triangleq \mathcal{B}^R(\mathcal{T}_2)$.
 - 3: $x = \text{Border Candidates Orientation}(\mathcal{A}', \mathcal{A}, \mathcal{B}, \mathcal{B}')$
 - 4: **if** $x = \text{correct}$ **then**
 - 5: Fix the root of \mathcal{T}_2 in α
 - 6: **else if** $x = \text{reverse}$ **then**
 - 7: Reverse the root of \mathcal{T}_2 , then fix it in α
 - 8: **else**
 - 9: Label \mathcal{T}_2 's root as nonorientable and continue the algorithm as if the root of \mathcal{T}_2 were fixed. This node is an actual Q -node of the tree of Robinson orderings.
 - 10: **end if**
 - 11: **Result:** The direct children of the root of \mathcal{T}_2 has been directly connected to α
-

Since $\mathcal{A}' \triangleq \mathcal{B}(\mathcal{T}_1)$, $\mathcal{B}' \triangleq \mathcal{B}(\mathcal{T}_3)$, $\mathcal{A} \triangleq \mathcal{B}^L(\mathcal{T}_2)$, and $\mathcal{B} \triangleq \mathcal{B}^R(\mathcal{T}_2)$ are border candidates for the arc $\partial\mathcal{T}_2$, the correctness of the procedure is due to the correctness of Algorithm 5.1. By Observation 5.11, the complexity is given by $\mathcal{O}(n \cdot \max\{|\mathcal{B}(\mathcal{T}_1)|, |\mathcal{B}(\mathcal{T}_2)|, |\mathcal{B}(\mathcal{T}_3)|\})$.

In a computational implementation of the algorithm, Q -nodes can be represented by tuples, whereas Q -trees are simply nested tuples. For clarity, we present an example of the Q -node fixing procedure mentioned in Algorithm 5.4.

In this example we consider Q -node $\alpha = ((0, 1, 2), (6, 5, (3, 4)), (7, (8, 9, 10)))$ taking values over $\mathcal{X} = [11]$. There are three Q -trees, which are consecutive in α , which are $\mathcal{T}_1 = (0, 1, 2)$, $\mathcal{T}_2 = (6, 5, (3, 4))$, and $\mathcal{T}_3 = (7, (8, 9, 10))$. An example of a possible outcome of Consecutive Orientation $(\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3)$ would be that the root of \mathcal{T}_2 is reversed and fixed into α . In such case α is modified into $((0, 1, 2), (3, 4), 5, 6, (7, (8, 9, 10)))$.

5.3.5. Complete internal orientation of a connected component and final orientation.

A complete internal orientation of a Q -tree \mathcal{T} is a process in which we determine the orientation of all nodes present in \mathcal{T} except from those present in \mathcal{T}^L and \mathcal{T}^R . For this task we propose the procedure Complete Internal Orientation (Algorithm 5.5).

Algorithm 5.5 Complete Internal Orientation

- 1: **Input:** A Q -tree \mathcal{T} with root α and $\text{depth}(\mathcal{T}) > 1$
 - 2: For $i \in [k]$, let $\alpha(i)$ be the i th children of α and let \mathcal{T}_i be the subtree whose root is $\alpha(i)$
 - 3: {Notice that $\mathcal{T}_0 = \mathcal{T}^L$ and $\mathcal{T}_{k-1} = \mathcal{T}^R$ }
 - 4: **while** $\exists 1 \leq j \leq k - 2$ such that the node $\alpha(j)$ has not been fixed into α **do**
 - 5: **for** $i = 1 \dots k - 2$ **do**
 - 6: Consecutive Orientation $(\mathcal{T}_{i-1}, \mathcal{T}_i, \mathcal{T}_{i-1})$
 - 7: **end for**
 - 8: **end while**
 - 9: **Result:** All elements in $\partial\mathcal{T} \setminus (\partial\mathcal{T}^L \cup \partial\mathcal{T}^R)$ are directly connected to α
-

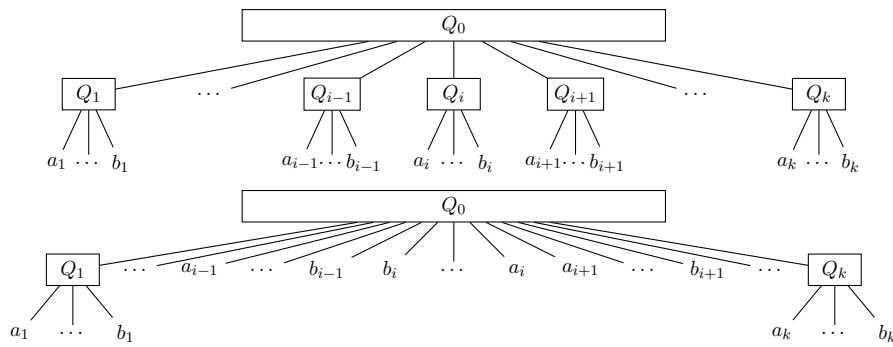


Figure 4. Example of a connected component of the nearest neighbors graph at the second recursion of *Recursive Seriation* before and after *Complete Internal Orientation*.

The orientation over all the nodes is done in a breadth-first search fashion.⁸ Once again, the correctness of the procedure is due to the correctness of [Algorithm 5.1](#). As an example, consider [Figure 4](#). Here, the tree at the top was constructed at the second recursion of the algorithm and represents a connected component of the nearest-neighbors graph over a family of Q -nodes. \mathcal{T}^L corresponds to the tree with root in Q_1 , and \mathcal{T}^R corresponds to the tree with root in Q_k . The tree in the bottom corresponds to the tree after [Algorithm 5.5](#).

In the final recursion of *Recursive Seriation* we obtain a unique connected component from *Arc Partition*, from which we construct a unique tree \mathcal{T} such that $\partial\mathcal{T} = \mathcal{X}$. Here, the cyclic order of \mathcal{X} implies that the subtrees \mathcal{T}^L and \mathcal{T}^R are consecutive. Hence, to orient these trees we make a slight variation in the procedure *Final Orientation* ([Algorithm 5.6](#)). This is equivalent to considering the Q -node as a ring rather than as a list.

Algorithm 5.6 Final Orientation

- 1: **Input:** A Q -tree \mathcal{T} with root α
 - 2: Let $\alpha(i)$ be the i th children of α and let \mathcal{T}_i be the subtree whose root is $\alpha(i)$
 - 3: **while** $\exists j \in [|\alpha|]$ such that the node $\alpha(j)$ has not been fixed into α **do**
 - 4: **if** $|\alpha| > 2$ **then**
 - 5: Let $\mathcal{T}_{-1} \triangleq \mathcal{T}^R$ and $\mathcal{T}_{|\alpha|} \triangleq \mathcal{T}^L$
 - 6: **for** $i \in [|\alpha|]$ **do**
 - 7: Consecutive Orientation($\mathcal{T}_{i-1}, \mathcal{T}_i, \mathcal{T}_{i+1}$)
 - 8: **end for**
 - 9: **else if** $|\alpha| = 2$ **then**
 - 10: Consecutive Orientation($\mathcal{T}_1^R, \mathcal{T}_0, \mathcal{T}_1^L$)
 - 11: Fix $\alpha(1)$ into α
 - 12: **end if**
 - 13: **end while**
 - 14: **Result:** All Q -nodes in \mathcal{T} are oriented
-

5.3.6. External orientation of trees. For each $\mathcal{T} \in \mathbf{T}$, the set $\partial\mathcal{T}$ corresponds to an arc, and as a consequence of [Lemma 5.14](#), $\mathbf{d}^{\min}(\mathcal{T}, \mathcal{T}')$ is attained at some $x \in \mathcal{B}(\mathcal{T})$ and $y \in \mathcal{B}(\mathcal{T}')$

⁸This way, trees of same depth are compared in [Algorithm 5.4](#) (excluding comparisons with \mathcal{T}^L and \mathcal{T}^R).

which are guaranteed to be borders of $\partial\mathcal{T}$ and $\partial\mathcal{T}'$, respectively. Therefore, we must arrange some of their internal nodes in such a way that x and y lie at the borders. With this purpose in mind, we propose the procedure **External Orientation** (Algorithm 5.7).

Algorithm 5.7 External Orientation

```

1: Input:  $Q$ -trees  $\mathcal{T}$  and  $\mathcal{T}'$ . The set  $\mathbf{d}^{\arg\min}(\mathcal{T}, \mathcal{T}')$ 
2: for  $(x, y) \in \mathbf{d}^{\arg\min}(\mathcal{T}, \mathcal{T}')$  do
3:   if  $x \in \mathcal{B}^L(\mathcal{T})$  then
4:     {Fix every  $Q$ -node in  $\mathcal{T}^L$  containing  $x$  as a descendant from the root until the  $Q$ -node  $\alpha$ 
      where  $x$  lies in a way such that  $x$  is placed on the left}
5:      $\mathcal{J} = \mathcal{T}^L$ 
6:     while  $\text{depth}(\mathcal{J}) > 0$  do
7:       if  $x \in \mathcal{B}^L(\mathcal{J})$  then
8:         Fix the root of  $\mathcal{J}$  into its parent
9:       else
10:        Reverse and then fix the root of  $\mathcal{J}$  into its parent
11:       end if
12:        $\mathcal{J} \leftarrow \mathcal{J}^L$ 
13:     end while
14:   else
15:     Proceed analogously, with  $\mathcal{B}^R(\mathcal{T})$  instead of  $\mathcal{B}^L(\mathcal{T})$ , and placing  $x$  at the right, instead of the
      left
16:   end if
17: end for
18: Complete Internal Orientation( $\mathcal{T}$ )
19: Repeat the same procedure with  $\mathcal{T}'$  and  $y$ 
20: Result: All  $Q$ -nodes in either  $\mathcal{T}^L$  or  $\mathcal{T}^R$  (resp.,  $\mathcal{T}^{L'}$  or  $\mathcal{T}^{R'}$ ) are oriented

```

An important observation is that in the tree \mathcal{T} resulting from the first part of this procedure we have that $\mathcal{T}^L = \{x\}$ (assuming for simplicity that $x \in \mathcal{B}^L(\mathcal{T})$). In the second part, we execute **Complete Internal Orientation** (Algorithm 5.5) with input \mathcal{T} . Since $\mathcal{T}^L = \{x\}$ at the end the only Q -nodes remaining to be oriented are the ones present in \mathcal{T}^R . As an example, we consider the Q -trees in Figure 5. Let \mathcal{T} be the tree in Figure 5(a). In this example, \mathcal{T}^L is the subtree with root in Q_1 , and \mathcal{T}^R is the singleton $\{b_0\}$. Suppose by computing $\mathbf{d}^{\min}(\mathcal{T}, \mathcal{T}')$ for some other \mathcal{T}' we get that \mathbf{d}^{\min} is attained at $b_2 \in \mathcal{B}^L(\mathcal{T})$. In that case, we must fix \mathcal{T}^L following the algorithm. Since b_2 is a left border in Q_1 , this node is correctly oriented. However, since b_2 appears in the right of Q_2 , we must reverse Q_2 as in Figure 5(b). The resulting tree is the one in Figure 5(c). Next, we perform a complete orientation, and the resulting tree is the one in Figure 5(d).

Notice that excluding the running time of Algorithm 5.5, the number of operations required for this procedure is bounded by $\mathcal{O}(\text{depth}(\mathcal{T}^L))$.

Observation 5.16. If \mathcal{T} is a tree built at the k th recursion of the algorithm, then clearly $\text{depth}(\mathcal{T}) \leq k$. We claim that since after this process either \mathcal{T}^L or \mathcal{T}^R gets completely oriented, it holds that $|\mathcal{B}(\mathcal{T})| \leq k + 1$. We prove this by induction on k . Notice that if \mathcal{T} is composed by a single Q -node in the root, then $|\mathcal{B}(\mathcal{T})| \leq 2$. Now let \mathcal{T} be a tree instantiated at the k th recursion of the algorithm. Without loss of generality (w.l.o.g.) assume \mathcal{T}^L gets completely

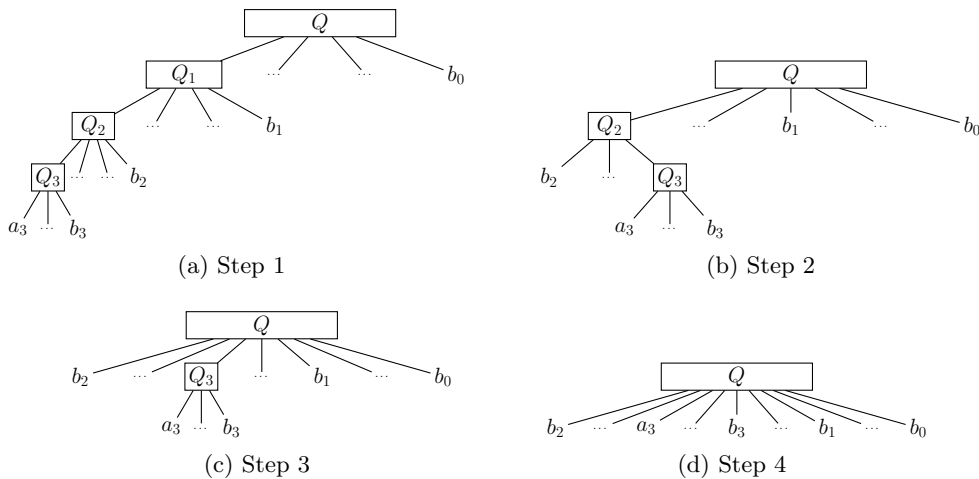


Figure 5. *External Orientation* over tree \mathcal{T} in which $\mathbf{d}^{\min}(\mathcal{T}, \mathcal{T}')$ is attained at $b_2 \in \mathcal{B}^L(\mathcal{T})$.

oriented. Then $\mathcal{B}(\mathcal{T}) = \mathcal{B}(\mathcal{T}^R) \cup \{x\}$. Hence, $|\mathcal{B}(\mathcal{T})| = |\mathcal{B}(\mathcal{T}^R)| + 1$. The claim follows by inducting on \mathcal{T}^R .

5.4. Analysis of the recursive seriation algorithm.

Theorem 5.17. *Given $D \in \text{pre-}\mathcal{C}_R^*$, let \mathcal{T} be the PQ-tree obtained from [Algorithm 5.2](#) with input D . Let $S(\mathcal{T})$ be the set of all orderings of \mathcal{X} (permutations) represented by the tree. Then, $S_{\mathcal{C}_R^*}(D) = \text{Dih}_n \circ S(\mathcal{T})$, i.e., it solves the strict circular seriation problem.*

Proof sketch. For simplicity, suppose we omit the orientation steps in [Algorithm 5.2](#) and leave them to the end of the process. This does not affect the set of solutions but may increase the time complexity. Denote by \mathcal{T}^{pre} and \mathcal{T} the trees before and after orientation, respectively. Also let \mathbf{T}_k be the family of trees instantiated at the k th recursive step. Notice that by [Lemma 5.13](#), evaluating \mathbf{d}^{\min} over \mathbf{T}_k yields a dissimilarity matrix $D_k \in \text{pre-}\mathcal{C}_R^*$. Due to [Proposition 5.3](#), we have that $S_{\mathcal{C}_R^*}(D) \subset \text{Dih}_n \circ S(\mathcal{T}^{\text{pre}})$ (at least all Robinson orderings are considered at this point). To complete the proof, it remains to show that in \mathcal{T} all orientable Q -nodes originally in \mathcal{T}^{pre} had been correctly fixed. To see this notice that the orientation of each Q -node in \mathcal{T}^{pre} is tested either by [External Orientation](#) ([Algorithm 5.7](#)) or [Consecutive Orientation](#) ([Algorithm 5.4](#)). The correctness of [External Orientation](#) is due to [Lemma 5.14](#). The correctness of [Consecutive Orientation](#) is due to [Corollary 5.10](#). ■

Theorem 5.18. *Recursive Seriation runs in $\mathcal{O}(n^2)$ time.*

Proof. We count the number of operations required by the procedure [Complete Internal Orientation](#) and [Final Orientation](#) separately from the rest. At the i th recursion let $\mathbf{T}(i)$ be the input Q -trees, let $k(i) \triangleq |\mathbf{T}(i)|$, and let $b(i) = \max_{\mathcal{T} \in \mathbf{T}(i)} |\mathcal{B}(\mathcal{T})|$. Then, by [Observation 5.15](#), computing \mathbf{d}^{\min} takes $\mathcal{O}(k(i)^2 \cdot b(i)^2)$ operations. By [Observation 5.16](#), the complexity of the procedure [External Orientation](#) takes $\mathcal{O}(k(i)^2)$ operations. Computing G_{NN} takes $\mathcal{O}(k(i)^2)$ operations. The procedure [Depth-First Search](#) takes $\mathcal{O}(k(i))$ operations.

On the other hand, notice that in each step of the recursion, every tree is merged to its nearest neighbor. This implies that $k(i) \leq \frac{n}{2^i}$ and, therefore, the depth of the recursion

is bounded by $\log_2(n)$. Since by [Observation 5.16](#) $b(i) \leq i + 1$, there is some constant $C_1 > 0$ such that the total number of operations of this procedure is bounded by $C_1 \sum_{i=0}^{\log_2(n)} \left(\frac{n}{2^i}\right)^2 (i+1)^2 + \left(\frac{n}{2^i}\right)^2 + \left(\frac{n}{2^i}\right) = \mathcal{O}(n^2)$.

It remains to consider **Complete Internal Orientation** and **Final Orientation**. In this procedures, all Q -nodes α are oriented through **Consecutive Orientation** ([Algorithm 5.4](#)) with input $(\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3)$ where α is the root of \mathcal{T}_2 . To count the operations of this procedure we consider two cases. The first (and most common) case is when $\mathcal{T}_1, \mathcal{T}_2$, and \mathcal{T}_3 are trees instantiated at the same recursive step. In this case, if they were instantiated at the i th recursion, then by [Observation 5.16](#) and [Observation 5.11](#) the orientation would take $\mathcal{O}((i+1) \cdot n)$ operations.

By counting on the recursion where each node was instantiated, the total number of operations involving first case Q -nodes can be bounded by $C_2 \cdot \sum_{i=0}^{\log_2(n)} \left(\frac{n}{2^i}\right) (i+1) \cdot n = \mathcal{O}(n^2)$.

A second case to consider is during the complete internal orientation of a connected component. Let \mathcal{T} be a tree generated from a connected component of the nearest-neighbors graph at the i th recursion of the algorithm. Then, in **Complete Internal Orientation** (or **Final Orientation**) with input \mathcal{T} , some of the internal Q -nodes will be oriented by having as border candidates $\mathcal{B}(\mathcal{T}^L)$ and $\mathcal{B}(\mathcal{T}^R)$. Since $\text{depth}(\mathcal{T}) \leq i$, this can occur for $\mathcal{B}(\mathcal{T}^L)$ (resp., $\mathcal{B}(\mathcal{T}^R)$) for at most i internal Q -nodes of \mathcal{T} . Let $C(i)$ be the number of connected components found in the i th recursion; then the number of second case Q -nodes is at most $C(i) \cdot i \cdot 2$. Since by [Observation 5.16](#), $|\mathcal{B}(\mathcal{T}^L)| \leq i$ and $|\mathcal{B}(\mathcal{T}^R)| \leq i$, the number of operations required for orienting all this nodes is bounded by $\mathcal{O}(C(i) \cdot i^2 \cdot n)$. Again, by counting through the recursion levels and considering that $C(i) \leq n/2^i$, the total cost of orienting second case Q -nodes is bounded by $C_3 \cdot \sum_{i=0}^{\log_2(n)} \left(\frac{n}{2^i}\right) \cdot i^2 \cdot n = \mathcal{O}(n^2)$, which proves the result. ■

5.5. PQ -tree of solutions in the strict Robinson case. It is clear that if a sequence is strictly monotone the only permutation that preserves this property is the one that reverses the sequence. Therefore if $D \in \mathcal{L}_R^*$, we have that $S_{\mathcal{L}_R^*}(D) = \{\mathbf{e}, \mathbf{r}\} \cong \text{Dih}_1$. However, it is not immediately clear which permutations are the ones that preserve strict unimodality. The next lemma will let us conclude that there is at most one *nontrivial* ordering for $D \in \mathcal{C}_R^*$.

Lemma 5.19. *Let $D \in \mathcal{C}_R^*$, and let $\mathcal{I}_1, \dots, \mathcal{I}_k$ be disjoint arcs of $[n]$. Let $\sigma_{\mathcal{I}_i}$ be the permutation that reverses \mathcal{I}_i . Then, at most one of the $\sigma_{\mathcal{I}_i}$'s produces a new Robinson ordering.*

Proof. Suppose $\sigma_{\mathcal{I}}$ is a Robinson ordering for some arc \mathcal{I} . For every $i \in [n]$, let $M(i) = \arg \max_j D(i, j)$. We claim that for every $i \notin \mathcal{I}$ it holds that $M(i) \subset \mathcal{I}$. Otherwise, given $m^* \in M(i)$, by the connectivity of \mathcal{I} , we must have that \mathcal{I} must be strictly contained in one of the two paths connecting i and m^* . Also notice that $D(i, \cdot)$ is strictly monotone in such a path. Hence, reversing \mathcal{I} would violate the monotonicity of such sequence (and thus the unimodality of the whole sequence). This proves the claim. Since \mathcal{I}^c is an arc, by the same argument we have that $i \in \mathcal{I}$ implies $M(i) \subset \mathcal{I}^c$. Hence, the only arcs that can be reversed are \mathcal{I} and \mathcal{I}^c . ■

6. Behavior for large n . The literature on the seriation problem has mostly focused on finite ordered sets, either linearly or cyclically ordered, on suitable classes of matrices encoding properties of this order, such as Robinson matrices, and on efficient algorithms for its solution. However, typically the use of seriation algorithms is motivated by the interpretation of data

as embedded in a closed curve, and it is unclear how these combinatorial solutions relate to the underlying order of a continuous object.

To bridge this gap, we provide a simple generative model of sampling from a continuous and periodic structure. That sample, and more specifically the dissimilarities between pairs of points from the sample, will be the input of our strict seriation algorithm. The question we want to answer is, *to what extent does the solution obtained by the seriation algorithm applied to a random sample reflect the underlying ordering of the periodic structure?* We will answer this question by proving that as the sample size n grows, the expected Kendall-tau distance from the strict circular seriation algorithm solution to the order inherited from the continuous model decreases at a rate $\mathcal{O}(\log(n)/n)$.

6.1. Reduction to \mathbb{S}^1 . We will consider our periodic continuous structure as parameterized by the unit circle. Equivalently, we will use the set $[0, 1)$ as the set of points, where we topologically identify 0 and 1, making it a circular-like structure. This set is endowed with the *natural cyclic order*, which results from embedding $[0, 1)$ into \mathbb{S}^1 .

We assume the set $[0, 1)$ is endowed with a dissimilarity \mathbf{d} . We will make some assumptions that relate the circular ordering to the circular Robinson property.

Assumption 6.1. \mathbf{d} is continuous and *strict circular Robinson*, i.e.,

$$(6.1) \quad \forall \text{ cyclically ordered } x, y, z, w \in [0, 1) : \mathbf{d}(y, w) > \min\{\mathbf{d}(y, x), \mathbf{d}(y, z)\}.$$

One natural question is how general this continuous model is. We claim that the assumption that our sample space is the unit circle is w.l.o.g. For example, if the sample space is a one dimensional compact manifold of \mathbb{R}^d , we can parameterize the manifold by its arc-length $\gamma : [0, 1) \mapsto \mathbb{R}^d$ and let $\mathbf{d}(t, s) := \|\gamma(t) - \gamma(s)\|$, which is clearly continuous. Notice, however, that the validity of the strict circular Robinson property is not guaranteed in this example: such an assumption depends on the relative positions of points in space.

6.2. Solutions in the limit. To understand the set of solutions in the limit we first need to characterize the natural symmetries of the strict Robinson dissimilarity \mathbf{d} . To do so, we consider the family of *cyclic shifts* $\{\pi_s : s \in [0, 1)\}$, defined by $\pi_s(t) = t + s \pmod{1}$, and the *reversal* $\pi_r(t) = 1 - t$. We let $\text{Dih}_\infty := \langle \pi_s, \pi_r : s \in [0, 1) \rangle$. In addition, given an arc $\mathcal{I} := (t, s) \subsetneq [0, 1)$, we let $\sigma_{\mathcal{I}}$ be the bijection that reverses \mathcal{I} and fixes \mathcal{I}^c . Since in the finite case all solutions can be expressed as compositions of such permutations, in the continuous case we look for solutions in $\text{Sym}(\infty) \triangleq \text{Dih}_\infty \circ \langle \sigma_{\mathcal{I}} : \mathcal{I} \text{ arc} \rangle$.

Theorem 6.2. *Suppose \mathbf{d} satisfies Assumption 6.1, and let $\pi \in \text{Sym}(\infty)$. If $\mathbf{d} \circ \pi$ is strict circular Robinson, then $\pi \in \text{Dih}_\infty$.*

This result can be seen as a well-posedness statement of the seriation problem in the continuous limit. Our next goal is to study its consequences for large (but finite) sample size.

6.3. Approximate well-posedness of seriation in the large n regime. We now propose a sampling model from the continuous model. We uniformly at random extract a size n sample from $[0, 1)$. We denote this sample by $\mathcal{X}_n := \{x_0, \dots, x_{n-1}\}$. If we let λ be the Lebesgue measure on $[0, 1)$, then our sampling is distributed as λ^n . Let $D_{\mathcal{X}_n}$ denote the

dissimilarity matrix associated to \mathcal{X}_n . In particular, if x_0, \dots, x_{n-1} are cyclically ordered, then the dissimilarity matrix is strict circular Robinson (cf. [Assumption 6.1](#)).

Although in the continuous case there is a unique Robinson ordering, with finitely many samples there might exist nontrivial orderings (cf. [Lemma 5.19](#)). In what follows we study conditions under which for a large sample, any ordering in $S_{C_R^*}(D\mathcal{X}_n)$ is close to the one induced by the curve. Our closeness measure is given by the Kendall-tau metric τ_K , and the goal is to bound the expected value of the diameter of the set of solutions:

Definition 6.3 (Kendall-tau metric [[13](#), [18](#)]). *We define the Kendall-tau distance between permutations π_1 and π_2 as $\tau_K(\pi_1, \pi_2) \triangleq |\mathcal{G}(\pi_1, \pi_2)| / \binom{n}{2}$, where $\mathcal{G}(\pi_1, \pi_2)$ corresponds to the set of discordant pairs defined as*

$$\mathcal{G}(\pi_1, \pi_2) \triangleq \{(i, j) : i < j, [\pi_1(i) < \pi_1(j) \wedge \pi_2(i) > \pi_2(j)] \vee [\pi_1(i) > \pi_1(j) \wedge \pi_2(i) < \pi_2(j)]\}.$$

The denominator $\binom{n}{2}$ ensures that $\tau_K(\pi_1, \pi_2) \in [0, 1]$. The next definition of diameter takes into account that for seriation cyclic permutations provide the same ordering.

Definition 6.4. *Given a set $S \subset \text{Sym}(n)$, the diameter of S is defined as $\text{diam}(S) \triangleq \max_{\pi_1, \pi_2 \in S} \min_{\hat{\pi}_1 \in \text{Dih}_n \circ \pi_1} \tau_K(\hat{\pi}_1, \pi_2)$.*

Let $\text{Arc} : [0, 1) \times [0, 1) \rightarrow [0, \frac{1}{2}]$ be the length of the shortest arc connecting two points in the unit circle, i.e., $\text{Arc}(\theta_1, \theta_2) = \min\{|\theta_1 - \theta_2|, 1 - |\theta_1 - \theta_2|\}$. To prove rates on the Kendall-tau distance we make a final assumption. This condition allows us to avoid making overly restrictive metric assumptions on the dissimilarity but still enjoy a weaker form of distance.

Assumption 6.5. The dissimilarity \mathbf{d} satisfies the following bi-Lipschitz property:

$$(6.2) \quad (\exists L \geq \ell > 0)(\forall s, t \in [0, 1)) \quad \ell \cdot \text{Arc}(s, t) \leq \mathbf{d}(s, t) \leq L \cdot \text{Arc}(s, t).$$

We conclude this section by providing a rate on the expected Kendall-tau diameter of the set of solutions of the circular Robinson algorithm. Hence, all these solutions must be close to the underlying order of the continuous model. Its proof is deferred to [Appendix A.4](#).

Theorem 6.6. *Let $\mathcal{X}_n = \{x_0, x_1, \dots, x_{n-1}\} \stackrel{iid}{\sim} \text{Unif}[0, 1)$. Then given any \mathbf{d} satisfying [Assumption 6.1](#) and [Assumption 6.5](#) we have that*

$$(6.3) \quad \mathbb{E}_{\mathcal{X}_n} \left[\text{diam}(S_{C_R^*}(D\mathcal{X}_n)) \right] = \mathcal{O}\left(\frac{(L+\ell)}{\ell} \cdot \frac{\log(n)}{n}\right).$$

Appendix A. Proofs.

A.1. Proof of Proposition 3.7. We need two auxiliary results first.

Proposition A.1. *f is unimodal (resp., strictly unimodal) if and only if for $i \leq j \leq k$ we have $f_j \geq \min\{f_i, f_k\}$ (resp., $f_j > \min\{f_i, f_k\}$).*

Proof. Suppose f is unimodal, let m be a mode, and suppose there are i, j, k , not all equal, such that $i \leq j \leq k$ and $x_j < \min\{x_i, x_k\}$. Then $x_i > x_j$ and $x_j < x_k$. This implies $m \leq j$ and $m \geq j$. Hence $m = j$. This is a contradiction. Now, suppose f satisfies the inequality but has no mode. Then j is not a mode, and there is $i < j$ and $k > j$ such that $f_i > f_j$ and $f_k > f_j$. This is a contradiction. The proof for the strictly unimodal case follows from the same arguments. ■

Proposition A.2. *If $(i_{-1}, i_0, i_1), (i_0, i_1, i_2) \in \mathcal{C}_n$, then for each $k \in \{-1, 1, 2\}$ there is $q_k \in [n]$ such that $i_k = i_0 + q_k \pmod n$. Furthermore, $q_1 \leq q_2 \leq q_{-1}$.*

Proof. Consider $q_k = i_k - i_0 \pmod n$. Then $q_0 = 0$. Since cyclic shifts do not change cyclic orderings, this implies $q_1 \leq q_2 \leq q_{-1}$. This proves the proposition. ■

Proof of Proposition 3.7. For simplicity we define $d_j^i \triangleq D(i, i + j \pmod n)$. ($2 \Rightarrow 1$) From Proposition A.2 we can write $i = j + q_i$, $k = j + q_k$, and $\ell = j + q_\ell$ with $q_k \leq q_\ell \leq q_i$. Since d^j is unimodal, from Proposition A.1 we deduce $d_{q_\ell}^j \geq \min\{d_{q_k}^j, d_{q_i}^j\}$. ($1 \Rightarrow 2$) If d^j is not unimodal, by Proposition A.1 there are $q_k \leq q_\ell \leq q_i$ with $d_{q_\ell}^j < d_{q_k}^j$ and $d_{q_\ell}^j < d_{q_i}^j$. If we define $i = j + q_i \pmod n$, $k = j + q_k \pmod n$, and $\ell = j + q_\ell \pmod n$ we see that $(i, j, k), (j, k, \ell) \in \mathcal{C}_n$. This contradicts 1. ■

A.2. Proofs for section 5.

Proof of Lemma 5.9. Let $z \in \mathcal{X}$, and denote $f_z(\cdot) \triangleq \mathbf{d}(z, \cdot)$. First, notice that

$$(A.1) \quad \begin{aligned} (\exists r > 0). \{a, a'\} \subset B_r(z) \wedge \{b, b'\} \subset B_r(z)^c \\ \Leftrightarrow \max\{f_z(a), f_z(a')\} < \min\{f_z(b), f_z(b')\}. \end{aligned}$$

($1 \Rightarrow 2$) By (A.1), this implication is direct from the fact that $a \in \mathcal{A}, b \in \mathcal{B}, a' \in \mathcal{A}'$ and $b' \in \mathcal{B}'$. ($2 \Rightarrow 1$) Let $r \triangleq \max\{f_z(x), f_z(x')\}$; thus $\{x, x'\} \subset B_r(z)$ and $\{y, y'\} \subset B_r(z)^c$. By Proposition 3.4 this ball is an arc, and therefore is connected in any Robinson ordering. This implies that all elements in between x and x' (in all Robinson orderings), including a and a' , must also be present in $B_r(z)$. Similarly, all elements in between y and y' , including b and b' , must not be present in $B_r(z)$. The implication follows from (A.1). ($3 \Rightarrow 2$) This follows directly. ($2 \Rightarrow 3$) Notice that given any $z \in \mathcal{X}$ and any $t > 0$ we have that if there is some $a \in \mathcal{A}$ and $a' \in \mathcal{A}'$ such that $\max\{f_z(a), f_z(a')\} < t$. Then, $f_z(a) < t \wedge f_z(a') < t$, which implies $\max\{\min f_z(\mathcal{A}), \min f_z(\mathcal{A}')\} < t$. Similarly, the existence of $b \in \mathcal{B}$ and $b' \in \mathcal{B}'$ such that $\min\{f_z(b), f_z(b')\} > t$ implies that $\min\{\max f_z(\mathcal{B}), \max f_z(\mathcal{B}')\} > t$. This proves the final implication and hence the result. ■

Proof of Lemma 5.13. We only prove the strict case, as the nonstrict case follows an analogous argument. Let $x_d \in B_d$ and $x_b \in B_b$ be such that $D^{\min}(B_b, B_d) = D(x_b, x_d)$, and let $x_c \in B_c, x_a \in B_a$ be arbitrary. We notice that x_a, x_b, x_c, x_d is cyclically ordered; hence

$$D^{\min}(B_b, B_d) = D(x_b, x_d) > \min\{D(x_b, x_a), D(x_b, x_c)\}.$$

On the other hand,

$$\min\{D(x_b, x_a), D(x_b, x_c)\} \geq \min\{D^{\min}(B_b, B_a), D^{\min}(B_b, B_c)\}$$

by definition of D^{\min} , proving the result. ■

A.3. Proof of Theorem 6.2.

Proof. Suppose by contradiction that there exist an arc $\mathcal{I} = [a, b)$ such that $\mathbf{d} \circ \sigma_{\mathcal{I}}$ is strict Robinson. For $\epsilon > 0$ small, $a - \epsilon, a, b, b + \epsilon$ are cyclically ordered. By hypothesis,

$$(A.2) \quad \mathbf{d}(\sigma(a), \sigma(b + \epsilon)) > \min\{\mathbf{d}(\sigma(a), \sigma(a - \epsilon)), \mathbf{d}(\sigma(a), \sigma(b))\},$$

and since $b + \epsilon, a - \epsilon \notin \mathcal{I}$, we get that $\sigma(a - \epsilon) = a - \epsilon$ and $\sigma(b + \epsilon) = b + \epsilon$. On the other hand, $\sigma(a) = b$ and $\sigma(b) = a$. Therefore, we can rewrite (A.2) as

$$(A.3) \quad \mathbf{d}(b, b + \epsilon) > \min\{\mathbf{d}(b, a - \epsilon), \mathbf{d}(b, a)\}.$$

Let $\delta := \mathbf{d}(b, a) > 0$. By continuity we get that $\mathbf{d}(\sigma(a), \sigma(b + \epsilon)) \rightarrow 0$ and $\mathbf{d}(b, a - \epsilon) \rightarrow \delta$ as $\epsilon \rightarrow 0$. For sufficiently small ϵ , this is a contradiction with (A.3). ■

A.4. Proof of Theorem 6.6. Given $x_0, \dots, x_{n-1} \in [0, 1)$, the *order statistics* correspond to the variables $x_{(1)}, x_{(2)}, \dots, x_{(k)}$ obtained by sorting the samples by increasing order. The *gaps* of the sample correspond to the variables $w_i \triangleq x_{(i+1)} - x_{(i)}$. Let $\epsilon_n \triangleq \max_{i \in [n]} w_i$. The following result can be found in [21, Theorem 1.2].

Proposition A.3. *Suppose $x_0, x_1, \dots, x_{n-1} \stackrel{iid}{\sim} \text{Unif}[0, 1)$. Then,*

$$\mathbb{P}(\epsilon_n \geq z) \leq \sum_{j=1}^{n+1} (-1)^{j-1} \binom{n+1}{j} (1 - jz)_+^n.$$

Proposition A.4. *Given any $\mathcal{I} = [x_i, x_j]$, we write $\mu(\mathcal{I})$ to denote $\text{Arc}(x_i, x_j)$. Suppose that Assumption 6.1 and Assumption 6.5 hold. Then the inequality $\epsilon_n < \ell\delta/(L + \ell)$ implies that any arc $\mathcal{I} \subset \mathcal{X}_n$ such that $\mu(\mathcal{I}) > \delta$ has a unique orientation in any circular Robinson ordering of $D\mathcal{X}_n$.*

Proof. Let $s \triangleq x_{(i)}$ and $t \triangleq x_{(j)}$ for some $i < j$. Let $\delta \in (0, \frac{1}{2})$, and consider the arc $\mathcal{I} = [s, t]$ in \mathcal{X}_n . Suppose $\epsilon_n < \ell\delta/(L + \ell)$ and $\mu(\mathcal{I}) > \delta$. Let $s^+ = x_{(i-1 \bmod n)}$ and $t^+ = x_{(j+1 \bmod n)}$. We claim that $B_s(\mathbf{d}(s, s^+)) \not\prec^* \mathcal{I}$. To prove the claim, it suffices to prove that

$$(A.4) \quad \mathbf{d}(s, s^+) < \min\{\mathbf{d}(s, t), \mathbf{d}(s, t^+)\}.$$

First, notice that $\mathbf{d}(s, s^+) \leq L \cdot \epsilon_n$. Second, notice that since $\text{Arc}(s, t) > \delta$, then

$$\mathbf{d}(s, t^+) \geq \ell \cdot \text{Arc}(s, t^+) \geq \ell \cdot (\text{Arc}(s, t) - \epsilon_n) \geq \ell \cdot (\delta - \epsilon_n),$$

and therefore $\min\{\mathbf{d}(s, t), \mathbf{d}(s, t^+)\} \geq \ell \cdot (\delta - \epsilon_n)$. Joining this two results with the fact that $\epsilon_n < \ell\delta/(L + \ell)$ proves the claim. ■

Lemma A.5. *Let $n \geq \log(1/\delta)/\delta$, and let $x_0, x_1, \dots, x_{n-1} \stackrel{iid}{\sim} \text{Unif}[0, 1)$. Let $E_n(\delta) = \{\epsilon_n < \ell\delta/(L + \ell)\}$; then*

$$\int_{E_n(\delta)} \text{diam } S_{\mathcal{C}_R^*}(D\mathcal{X}_n(\omega)) \, d\lambda^n(\omega) = \mathcal{O}\left(\delta^2 + \frac{\delta \log(1/\delta)}{n}\right).$$

Proof. Recall from Lemma 5.19 that there is at most one nontrivial ordering $\sigma_{\mathcal{I}}$ of $D\mathcal{X}_n$ which corresponds to the permutation that reverses the arc $\mathcal{I} \cap \mathcal{X}_n$. Therefore, it suffices to bound the integral of the random variable $\tau_K(\text{id}, \sigma^*)$, where $\sigma^* \in \arg \min_{\hat{\sigma} \in \{\pi_r \circ \sigma_{\mathcal{I}}, \sigma_{\mathcal{I}}\}} \tau_K(\text{id}, \hat{\sigma})$.

The number of discordant pairs between σ^* and id is bounded by $\frac{1}{2} \min\{|\mathcal{I} \cap \mathcal{X}_n|, n - |\mathcal{I} \cap \mathcal{X}_n|\}^2$. Hence, we will focus on bounding this expression. Let $\omega \in E_n(\delta)$. By Proposition A.4, $\mu(\mathcal{I}(\omega)) \leq \delta$. This implies that either $\lambda(\mathcal{I}) \leq \delta$ or $\lambda(\mathcal{I}) \geq 1 - \delta$. Therefore,

$$\int_{E_n(\delta)} \min\{|\mathcal{I} \cap \mathcal{X}_n|, n - |\mathcal{I} \cap \mathcal{X}_n|\}^2 d\lambda^n(\omega) \\ \leq \int_{E_n(\delta)} \max_{\mathcal{J} \text{ interval}, \lambda(\mathcal{J}) \leq \delta} |\mathcal{J} \cap \mathcal{X}_n|^2 d\lambda^n(\omega) \leq \int_{\Omega} \max_{\mathcal{J} \text{ interval}, \lambda(\mathcal{J}) \leq \delta} |\mathcal{J} \cap \mathcal{X}_n|^2 d\lambda^n(\omega).$$

We bound the random variable inside the integral using a balls and bins argument. W.l.o.g. $1/\delta$ is an integer. Let $(\mathcal{J}_i)_{i \in [1/\delta]}$ be a partition of $[0, 1]$ by disjoint intervals of length δ . For any $\omega \in \Omega$, the maximizer in the integral above lies in at most two of the partition intervals. Therefore, $\max_{\lambda(\mathcal{J}) \leq \delta} |\mathcal{J} \cap \mathcal{X}_n| \leq 2 \max_{i \in [n]} |\mathcal{J}_i \cap \mathcal{X}_n|$. Next, we can estimate $\max_{i \in [n]} |\mathcal{J}_i \cap \mathcal{X}_n|$ by looking into the problem of throwing n balls into $1/\delta$ bins (see [24] for further details); since we further assumed that $n \geq 1/\delta \log(1/\delta)$, then with high probability, the maximum occupancy is bounded by $n\delta + \Theta(\sqrt{n\delta \log(1/\delta)})$. Plugging this bound above yields the result. ■

Proof of Theorem 6.6. Let $E_n(\delta)$ denote the event $\{\epsilon_n < \ell\delta/(L + \ell)\}$. Denote the random variable $Z = \text{diam}(S_{C_R^*}(D\mathcal{X}_n))$. Then,

$$(A.5) \quad \mathbb{E}[Z] = \int_{E_n(\delta)} Z(\omega) d\lambda(\omega) + \int_{E_n(\delta)^c} Z(\omega) d\lambda(\omega) \leq \mathcal{O}\left(\delta^2 + \frac{\delta \log(1/\delta)}{n}\right) + \mathbb{P}[E_n(\delta)^c],$$

where in the inequality we used Lemma A.5 and $\tau_K \leq 1$. By Proposition A.3 we have

$$(A.6) \quad \mathbb{P}[E_n(\delta)^c] \leq \sum_{j=1}^{n+1} (-1)^{j-1} \binom{n+1}{j} (1 - jx)_+^n \leq \sum_{j=1}^{n+1} \left(\frac{e(n+1)}{j}\right)^j \exp\{-xjn\},$$

where $x = \ell\delta/(L + \ell)$. By taking $\delta(n) = (L + \ell) \cdot \log(e(n + 1)^2)/(n \cdot \ell)$ we obtain that (A.6) can be bounded by $\sum_{j=1}^{n+1} \frac{1}{(n+1)^j} \in \mathcal{O}(1/n)$. By (A.5) and (A.6) we conclude that $\mathbb{E}[Z] = \mathcal{O}\left(\frac{L+\ell}{\ell} \frac{\log n}{n}\right)$. ■

Appendix B. Auxiliary subroutines.

Algorithm B.1 Arc Partition

- 1: **Input:** A dissimilarity \mathbf{d} and a set \mathbf{T}
 - 2: $\mathcal{N}_G(x) \triangleq \{y \in \mathbf{T} : x \in \text{NN}(y) \vee y \in \text{NN}(x)\}$ {Compute the neighborhood function}
 - 3: $\mathcal{B} \triangleq \{x \in \mathbf{T} : |\mathcal{N}_G(x)| = 1\}$
 {Find all degree 1 nodes (if there are no such nodes pick any)}
 - 4: $i = 0$ {Run DFS starting at every nonvisited degree 1 node}
 - 5: **for** $x \in \mathcal{B} \setminus \cup_{j < i} \alpha_j$ **do**
 - 6: $\alpha_i = \text{DFS}(\mathcal{N}_G, \emptyset, x)$
 - 7: $i = i + 1$
 - 8: **end for**
 - 9: **Output:** An arc partition stored into tuples $\mathcal{P} \triangleq \{\alpha_i\}_{i \in [k]}$
-

Algorithm B.2 Depth-First Search

- 1: **Input:** The neighborhood function of a graph $\mathcal{N}_G(\cdot)$, a tuple α of visited nodes and a starting node x
 - 2: $\alpha(n) = x$ {Set x as n th visited node where n is the size of α }
 - 3: **for** $y \in \mathcal{N}_G(x) \setminus \alpha$ **do**
 - 4: $\alpha = \text{DFS}(\mathcal{N}_G, \alpha, y)$ {Recurse over all adjacent nodes that have not been visited}
 - 5: **end for**
 - 6: **return** α
 - 7: **Output:** A tuple of visited nodes α
-

Acknowledgment. We would like to thank Alexandre d’Aspremont for valuable discussions at different stages of this work.

REFERENCES

- [1] J. E. ATKINS, E. G. BOMAN, AND B. HENDRICKSON, *A spectral algorithm for seriation and the consecutive ones problem*, SIAM J. Comput., 28 (1998), pp. 297–310.
- [2] M. BELKIN AND P. NIYOGI, *Laplacian eigenmaps for dimensionality reduction and data representation*, Neural Comput., 15 (2003), pp. 1373–1396.
- [3] K. S. BOOTH AND G. S. LUEKER, *Linear algorithms to recognize interval graphs and test for the consecutive ones property*, in Proceedings of the Seventh Annual ACM Symposium on Theory of Computing, 1975, pp. 255–265.
- [4] K. S. BOOTH AND G. S. LUEKER, *Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms*, J. Comput. Syst. Sci., 13 (1976), pp. 335–379.
- [5] F. BRUCKER AND C. OSSWALD, *Hypercycles and Dissimilarities*, manuscript, 2008.
- [6] V. CHEPOI AND B. FICHET, *Recognition of Robinsonian dissimilarities*, J. Classification, 14 (1997), pp. 311–325.
- [7] V. CHEPOI, B. FICHET, AND M. SESTON, *Seriation in the presence of errors: NP-hardness of ℓ^∞ -fitting Robinson structures to dissimilarity matrices*, J. Classification, 26 (2009), pp. 279–296.
- [8] V. CHEPOI AND M. SESTON, *Seriation in the presence of errors: A factor 16 approximation algorithm for ℓ^∞ -fitting Robinson structures to distances*, Algorithmica, 59 (2011), pp. 521–568.
- [9] R. R. COIFMAN, Y. SHKOLNISKY, F. J. SIGWORTH, AND A. SINGER, *Graph Laplacian tomography from unknown random projections*, IEEE Trans. Image Process., 17 (2008), pp. 1891–1899.
- [10] X. EVANGELOPOULOS, A. J. BROCKMEIER, T. MU, AND J. Y. GOULERMAS, *Circular object arrangement using spherical embeddings*, Pattern Recogn., 103 (2020), 107192.
- [11] D. FULKERSON AND O. GROSS, *Incidence matrices and interval graphs*, Pacific J. Math., 15 (1965), pp. 835–855.
- [12] L. HUBERT, P. ARABIE, AND J. MEULMAN, *Graph-theoretic representations for proximity matrices through strongly-anti-Robinson or circular strongly-anti-Robinson matrices*, Psychometrika, 63 (1998), pp. 341–358.
- [13] M. G. KENDALL, *A new measure of rank correlation*, Biometrika, 30 (1938), pp. 81–93.
- [14] J. KÖBLER, S. KUHNERT, AND O. VERBITSKY, *Circular-arc hypergraphs: Rigidity via connectedness*, Discrete Appl. Math., 217 (2017), pp. 220–228.
- [15] M. LAURENT AND M. SEMINAROTI, *Similarity-first search: A new algorithm with application to Robinsonian matrix recognition*, SIAM J. Discrete Math., 31 (2017), pp. 1765–1800.
- [16] Y.-C. LIAO, H.-W. CHENG, H.-C. WU, S.-C. KUO, T.-L. LAUDERDALE, AND F.-J. CHEN, *Completing circular bacterial genomes with assembly complexity by using a sampling strategy from a single minion run with barcoding*, Front. Microbiol., 10 (2019), 2068.
- [17] I. LIIV, *Seriation and matrix reordering methods: An historical overview*, Stat. Anal. Data Min., 3 (2010), pp. 70–91.
- [18] R. MA, T. TONY CAI, AND H. LI, *Optimal permutation recovery in permuted monotone matrix model*, J. Amer. Statist. Assoc., 116 (2021), pp. 1358–1372.
- [19] B. G. MIRKIN AND S. N. RODIN, *Graphs and Genes*, Springer, Cham, 1984.
- [20] V. NOVÁK, *Cyclically ordered sets*, Czechoslovak Math. J., 32 (1982), pp. 460–473.
- [21] I. PINELIS, *Order Statistics on the Spacings Between Order Statistics for the Uniform Distribution*, arXiv preprint, [arXiv:1909.06406](https://arxiv.org/abs/1909.06406) [math.ST], 2019.
- [22] P. PRÉA AND D. FORTIN, *An optimal algorithm to recognize Robinsonian dissimilarities*, J. Classification, 31 (2014), pp. 351–385.
- [23] A. QUILLIOT, *Circular representation problem on hypergraphs*, Discrete Math., 51 (1984), pp. 251–264.
- [24] M. RAAB AND A. STEGER, *“Balls into bins”—a simple and tight analysis*, in Proceedings of the International Workshop on Randomization and Approximation Techniques in Computer Science, Springer, Cham, 1998, pp. 159–170.

- [25] A. RECANATI, T. BRÜLS, AND A. D'ASPREMONT, *A spectral algorithm for fast de novo layout of uncorrected long nanopore reads*, *Bioinformatics*, 33 (2017), pp. 3188–3194.
- [26] A. RECANATI, T. KERDREUX, AND A. D'ASPREMONT, *Reconstructing Latent Orderings by Spectral Clustering*, arXiv preprint, [arXiv:1807.07122](https://arxiv.org/abs/1807.07122) [cs.DS], 2018.
- [27] W. S. ROBINSON, *A method for chronologically ordering archaeological deposits*, *Amer. Antiquity*, 16 (1951), pp. 293–301.
- [28] M. SESTON, *Dissimilarités de Robinson: Algorithmes de Reconnaissance et d'Approximation*, PhD thesis, Aix-Marseille 2, 2008.
- [29] A. TUCKER, *Matrix characterizations of circular-arc graphs*, *Pacific J. Math.*, 39 (1971), pp. 535–545.